

# 低轨道卫星功率带宽资源联合分配方法

徐双<sup>1</sup>, 王兴伟<sup>2</sup>, 黄敏<sup>3</sup>

(1. 东北大学 计算机科学与工程学院, 辽宁 沈阳 110169; 2. 东北大学 软件学院, 辽宁 沈阳 110169;  
3. 东北大学 信息科学与工程学院, 辽宁 沈阳 110819)

**摘 要:** 由于低轨道卫星上功率和带宽资源有限,设计有效的资源分配方法具有重要的意义. 基于此提出了一种低轨道星间功率带宽资源联合分配方法,提高了星间资源分配的公平性和网络业务承载能力. 该方法应用地面站数目和星地链路存在时间刻画星间链路容量的比例,将星间资源分配问题归纳为非线性混合整数规划问题. 通过引入动态可行域,定义二元变量更新操作来改进燕子群算法,用于优化模型求解. 仿真得出卫星瞬时吞吐容量和瞬时网络容量变化具有周期性,并伴有随机波动,所提方法牺牲了网络容量,获取了星间资源分配的公平性.

**关 键 词:** 低轨道卫星;功率;带宽;联合分配;燕子群优化

中图分类号: TP 393      文献标志码: A      文章编号: 1005-3026(2017)03-0320-06

## Power and Bandwidth Joint Allocation Method for LEO Satellites

XU Shuang<sup>1</sup>, WANG Xing-wei<sup>2</sup>, HUANG Min<sup>3</sup>

(1. School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China; 2. School of Software, Northeastern University, Shenyang 110169, China; 3. School of Information Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: WANG Xing-wei, E-mail: wangxw@mail.neu.edu.cn)

**Abstract:** Onboard power and bandwidth of low earth orbit (LEO) satellite networks are limited, thus designing an efficient resource allocation method is of great significance. A power and bandwidth joint allocation method for LEO satellites was proposed, and improved the resource allocation fairness among satellites and network service carrying ability were improved. The ground stations number and user data links life time were used to describe proportional constraints of intersatellite links capacity, and satellite resources allocation problem was formulated as a nonlinear mixed integer programming problem. Swallow swarm optimization (SSO) was improved by introducing dynamic feasible region and defining binary variable update operation, then the optimization model was solved by using this method. Simulations demonstrate satellite instantaneous throughput capacity and network instantaneous capacity change periodically with random fluctuations. The method achieves the resource allocation fairness among satellites at cost of network capacity.

**Key words:** low earth orbit (LEO) satellites; power; bandwidth; joint allocation; swallow swarm optimization

低轨道卫星网络由 LEO 卫星组成,卫星运行速度快,拓扑频繁变化,星地往返延迟小,对天线尺寸和传输功率要求低,在快接入、广覆盖,以及

高带宽实时服务中发挥着重要的作用,但 LEO 卫星上资源有限,设计有效的星间网络资源分配算法,有利于提高网络的业务承载能力和资源利

用率。

文献[1]设计了保证卫星资源有效利用的带宽和功率分配算法。文献[2]考虑 QoS 需求,用拉格朗日启发算法求解低轨道卫星网络子载波分配问题。文献[3]考虑延迟对实时业务的影响,提出了一种功率和带宽资源分配方法,实现了吞吐量和延迟间的平衡。文献[4]提出了一种多波束卫星网络的功率和带宽分配方法,提高了波束间容量的公平性。上述方法主要面向下行链路资源,不适用于星间链路。

本文提出一种面向网络容量的低轨道星间功率和带宽资源联合分配方法。首先,将带宽分配转化为子信道分配,将低轨道星间资源分配问题归纳为一个以最大化卫星节点吞吐容量为目标的非线性混合整数规划问题。其次,综合考虑星地链路存在时间和卫星覆盖的地面站数目,刻画相邻卫星间星间链路容量比例公平性约束。最后,受智能优化方法在此类问题求解中成功应用<sup>[5-7]</sup>的启发,本文引入动态可行域,定义二元参数更新操作,采用燕子群优化(swallow swarm optimization, SSO)算法<sup>[8-9]</sup>求解此问题。在铱星和全球星系统上仿真该算法,并用4个不同指标评价该算法性能。

## 1 问题描述

### 1.1 低轨道卫星网络

构成低轨道卫星网络的 LEO 卫星之间凭借星间链路互相通信,网络拓扑周期性变化。链路距离的变化和通断,造成网络容量波动。每颗 LEO 卫星与同一轨道上前两相邻卫星间建立的持续连接链路称为同轨链路,与左右轨道上的两卫星建立的链路称为异轨链路,随卫星的绕行断开或重连。

### 1.2 容量分析

假设卫星  $s_{i_0j_0}$  总带宽为  $B$ , 相邻卫星数为  $N_{nei}$ 。将  $s_{i_0j_0}$  总带宽均分给  $N_{sub}$  个正交子信道,通过分配子信道来分配带宽资源。子信道分配结果  $\mathbf{C}$  表示为  $\mathbf{C} = [c_{1,1}, c_{1,2}, \dots, c_{1,N_{sub}}, c_{2,1}, c_{2,2}, \dots, c_{2,N_{sub}}, \dots, c_{N_{nei},1}, c_{N_{nei},2}, \dots, c_{N_{nei},N_{sub}}]$ 。其中,  $c_{k,i} \in \{0,1\}$ ,  $k \in [1, N_{nei}]$ ,  $i \in [1, N_{sub}]$ ,  $c_{k,i} = 1$  表示第  $i$  个子信道被分配给第  $k$  颗相邻卫星。功率分配结果为  $\mathbf{P} = [p_1, p_2, \dots, p_{N_{sub}}]$ , 且  $\sum_{i=1}^{N_{sub}} p_i \leq p_{total}$ ,  $p_i$  为分配给子信道  $i$  的发射功率,  $p_{total}$  为  $s_{i_0j_0}$  的总功率。

定义  $t$  时刻  $s_{i_0j_0}$  与第  $k$  个相邻卫星间的链路容

量  $T_k^{link}$  为  $s_{i_0j_0}$  分配给该相邻卫星的子信道容量和:

$$T_k^{link} = \frac{B}{N_{sub}} \sum_{i=1}^{N_{sub}} c_{k,i} \text{lb}(1 + \text{SNR}_{k,i}) \quad (1)$$

其中,  $\text{SNR}_{k,i}$  为第  $k$  个相邻卫星的第  $i$  个子信道的信噪比。

$$s_{i_0j_0} \text{ 的瞬时吞吐容量 } \text{SITC}_{i_0j_0}(t) = \sum_{k=1}^{N_{nei}} T_k^{link}.$$

根据拓扑中链路是否发生切换,将网络周期  $T$  划分成若干时间片,每个时间片内,网络容量较稳定。每个时间片内设置若干采样点,应用采样点容量近似计算  $T$  内的平均吞吐容量  $\text{SATC}_{i_0j_0}$ , 如式(2)所示:

$$\text{SATC}_{i_0j_0} = \frac{1}{|S_{\text{sample}}|} \sum_{t_i \in S_{\text{sample}}} \text{SITC}_{i_0j_0}(t_i) \quad (2)$$

其中:  $S_{\text{sample}}$  为采样时刻集合;  $|S_{\text{sample}}|$  为采样点数。

$t$  时刻的瞬时网络容量  $\text{INC}(t) = \sum_{i=1}^{|S|} \text{SITC}_i(t)$ , 其中  $|S|$  为卫星数目。周期  $T$  内的平均网络容量  $\text{ANC} = \frac{1}{|S_{\text{sample}}|} \sum_{t_i \in S_{\text{sample}}} \text{INC}(t_i)$ 。

### 1.3 链路存在时间

地面用户通过地面站接入卫星系统,而卫星不断绕地球转动,故不同时刻同一卫星覆盖的地面站不断变化。假设当前时刻  $t_{\text{now}}$  地面站  $G_0$  位于卫星  $S_0$  的覆盖范围内,时刻  $t_{\text{end}}$  时  $S_0$  恰好不再覆盖  $G_0$ , 则定义时间长度  $t_{\text{end}} - t_{\text{now}}$  为星地链路存在时间  $T_{\text{lfp}}$ 。假设  $t_{\text{begin}}$  时刻  $G_0$  恰好进入  $S_0$  的覆盖范围,则定义时间长度  $t_{\text{end}} - t_{\text{begin}}$  为星地链路最大存在时间  $T_{\text{mlfp}}$ 。

### 1.4 问题建模

同一卫星可能同时覆盖多个地面站,并与这些地面站同时通信。所以卫星覆盖的地面站数越多,卫星传输的业务量越大,需要分配的资源越多。此外,星地链路存在时间越长,被用来传输数据的概率越大,所需分配的资源越多。本文采用卫星覆盖的地面站数和星地链路存在时间刻画比例系数,从而表征相邻卫星间链路容量的比例约束,提高资源分配中各卫星间的公平性。以最大化节点吞吐量为目标,建立带宽功率联合优化模型如下:

$$\max \quad \text{SITC}_{i_0j_0} \quad (3)$$

$$\text{s. t.} \quad \sum_{i=1}^{N_{sub}} p_i \leq p_{total}; \quad (4)$$

$$p_i \in [0, p_{total}], i = 1, \dots, N_{sub}; \quad (5)$$

$$\text{SNR}_i \geq \text{SNR}_{\min}, i = 1, \dots, N_{sub}; \quad (6)$$

$$\sum_{k=1}^{N_{nei}} c_{k,i} = 1, i = 1, \dots, N_{sub}; \quad (7)$$

$$c_{k,i} \in \{0,1\}, k=1,\dots,N_{\text{nei}}; i=1,\dots,N_{\text{sub}}; \quad (8)$$

$$T_k^{\text{link}}: T_l^{\text{link}} = \varphi_k \cdot \varphi_l, \forall k, l \in \{1, \dots, N_{\text{nei}}\}, k \neq l. \quad (9)$$

式(6)为星际链路正常通信的信噪比约束,式(9)为相邻卫星间星际链路的容量比例约束,比例系数  $\varphi_k$  的计算如式(10)所示:

$$\varphi_k = \beta_{\text{gst}} \cdot \sum_{i \in \Omega_G} \frac{T_{\text{lp}}^{k,i}}{T_{\text{mlfp}}}, \quad k=1,2,\dots,N_{\text{nei}}. \quad (10)$$

其中:  $\beta_{\text{gst}}$  为地面站对链路容量的影响因子;集合  $\Omega_G$  表示  $s_{i_0}$  的第  $k$  颗相邻卫星覆盖的地面站。

## 2 方法设计

上述模型属于非线性混合整数规划问题,本文采用 SSO 算法求解。

### 2.1 动态可行域和适应度函数

SSO 算法寻优中,可行解和非可行解并存,而进化后期种群中基本都是可行解,但这些可行解的质量可能很差,导致最终收敛到局部最优解。可行域附近的非可行解违反约束程度低,目标函数值接近最优值,能为搜索过程提供较多信息。故本文引入动态可行域,即由可行域和可行域附近的部分非可行解组成的集合,使得搜索过程从可行解和非可行解两个方向同时逼近最优解。

动态可行域的设置需考虑非可行解的个数和选择准则。迭代初期,选择较多的非可行解组成动态可行域,可以增加种群的多样性,但由于约束问题的最优解必须是可行解,所以随着迭代次数的增加,非可行解的个数逐渐减小,使动态可行域逐渐收缩到实际可行域。非可行解的个数  $N_{\text{select}}$  变化如下:

$$N_{\text{select}} = N_{\text{lea}} \cdot \left(1 - \frac{\text{iter}}{\text{maxIter}}\right). \quad (11)$$

其中:  $N_{\text{lea}}$  为领队燕子数(详见 2.3 节); iter 为当前迭代次数; maxIter 为最大迭代次数。

将优化模型中的约束整理为不等式约束  $g_1(\mathbf{X})$  和等式约束  $g_2(\mathbf{X})$ , 构造违反约束程度  $\text{Con}(\mathbf{X})$ , 如式(12)~式(14)所示:

$$g_1(\mathbf{X}) = \left(\sum_{i=1}^{N_{\text{sub}}} p_i - p_{\text{total}}\right) + \sum_{i=1}^{N_{\text{sub}}} (\text{SNR}_{\text{min}} - \text{SNR}_i) \leq 0; \quad (12)$$

$$g_2(\mathbf{X}) = \sum_{i=1}^{N_{\text{sub}}} \left(\sum_{k=1}^{N_{\text{nei}}} c_{k,i} - 1\right) + \sum_{k \neq l}^{N_{\text{nei}}} \left(\frac{T_k^{\text{link}}}{T_l^{\text{link}}} - \frac{\varphi_k}{\varphi_l}\right) = 0; \quad (13)$$

$$\text{Con}(\mathbf{X}) = (g_1(\mathbf{X}))^2 + (g_2(\mathbf{X}))^2. \quad (14)$$

构造目标函数  $\text{Obj}(\mathbf{X}) = \text{SITC}_{i_0}$ , 并归一化为  $F(\mathbf{X}) = \frac{\text{Obj}(\mathbf{X}) - \text{Obj}_{\text{min}}}{\text{Obj}_{\text{max}} - \text{Obj}_{\text{min}}}$ , 用  $G(\mathbf{X}) =$

$\frac{\text{Con}_{\text{max}} - \text{Con}(\mathbf{X})}{\text{Con}_{\text{max}}}$  归一化违反约束程度。

根据  $F(\mathbf{X})$  和  $G(\mathbf{X})$  构造评判函数  $\text{Est}(\mathbf{X}_{\text{inf}})$  为  $\text{Est}(\mathbf{X}_{\text{inf}}) = F(\mathbf{X}_{\text{inf}}) \cdot (G(\mathbf{X}_{\text{inf}}))^{(0.5 + \text{iter}/\text{maxIter})}$ . (15)

计算各非可行解的评判函数值并排序,选取前  $N_{\text{select}}$  个非可行解组成动态可行域  $D_{\text{fea}}$ .

根据动态可行域构造适应度函数:

$$\text{Fit}(\mathbf{X}) = \begin{cases} F(\mathbf{X}), & \mathbf{X} \in D_{\text{fea}}; \\ G(\mathbf{X}) - 1, & \mathbf{X} \notin D_{\text{fea}}. \end{cases} \quad (16)$$

### 2.2 解的初始化

燕子的位置对应优化问题中的解,每个解  $\mathbf{X} = (\mathbf{C}, \mathbf{P})$  由整数向量  $\mathbf{C}$  和实数向量  $\mathbf{P}$  构成,根据式(17)和式(18)初始化  $\mathbf{C}$  和  $\mathbf{P}$ . 燕子飞行速度分解为沿主要领队燕子方向的速度  $\mathbf{v}_{\text{HL}}$  和沿局部领队燕子方向的速度  $\mathbf{v}_{\text{LL}}$ , 初始化方式与解的初始化方式相同。

$$c_l = \text{rand} \{0,1\}, l=1,2,\dots,N_{\text{nei}} \cdot N_{\text{sub}}; \quad (17)$$

$$p_m = \text{rand} \cdot p_{\text{total}}, m=1,2,\dots,N_{\text{sub}}. \quad (18)$$

### 2.3 领队燕子

领队燕子包括主要领队燕子和局部领队燕子。SSO 算法将整个燕子种群分成多个部分,每个部分有 1 只局部领队燕子,整个种群有 1 只主要领队燕子,引导整个燕子群的运动方向,其他燕子的速度主要受二者影响。根据燕子适应度值对所有燕子进行排序,把适应度值最大的燕子设为主要领队燕子,再按照排序依次选取  $N_{\text{lea}} - 1$  个燕子作为局部领队燕子。在迭代的过程中,根据上述原则重新排序所有燕子,选取新的主要领队燕子和局部领队燕子。

### 2.4 探索燕子

探索燕子是燕子群中的主要部分,负责搜索状态空间。探索燕子根据飞行速度  $\mathbf{v}^{\text{new}}$  飞到新位置  $\mathbf{X}^{\text{new}}$ ,  $\mathbf{v}^{\text{new}}$  沿主要领队燕子方向的速度为  $\mathbf{v}_{\text{HL}}^{\text{new}}$  和沿局部领队燕子方向的速度为  $\mathbf{v}_{\text{LL}}^{\text{new}}$ .

$$\mathbf{v}_{\text{HL}}^{\text{new}} \text{ 和 } \mathbf{v}_{\text{LL}}^{\text{new}} \text{ 的实数向量 } \mathbf{v}_{\text{PHL}}^{\text{new}} \text{ 和 } \mathbf{v}_{\text{PLL}}^{\text{new}} \text{ 更新方式为} \\ \mathbf{v}_{\text{P}}^{\text{new}} = \mathbf{v}_{\text{P}}^{\text{old}} + \alpha \cdot \text{rand}() \cdot (\mathbf{X}_{\text{P}}^{\text{His}} - \mathbf{X}_{\text{P}}^{\text{old}}) + \\ \beta \cdot \text{rand}() \cdot (\mathbf{X}_{\text{P}} - \mathbf{X}_{\text{P}}^{\text{old}}). \quad (19)$$

其中:  $\mathbf{v}_{\text{P}}^{\text{old}}$  是探索燕子当前速度沿领队燕子方向的实数部分;  $\text{rand}()$  是  $[0,1]$  内的随机数;  $\mathbf{X}_{\text{P}}^{\text{His}}$  是该探索燕子历史最优位置的实数部分;  $\mathbf{X}_{\text{P}}$  是领队燕子位置的实数部分;  $\mathbf{X}_{\text{P}}^{\text{old}}$  是探索燕子当前位置的实数部分;  $\alpha$  和  $\beta$  是探索燕子沿领队燕子方向速度收敛参数,调整方式如下:

$$\alpha = \left(1 - \frac{\text{Fit}(\mathbf{X}^{\text{old}}) + 1}{\text{Fit}(\mathbf{X}) + 1}\right) \cdot [(a_1 - a_2) \cdot \frac{\text{iter}}{\text{maxIter}} + a_2]; \quad (20)$$

$$\beta = (1 - \frac{\text{Fit}(\boldsymbol{X}^{\text{old}}) + 1}{\text{Fit}(\boldsymbol{X}) + 1}) \cdot [(a_2 - a_1) \cdot \frac{\text{iter}}{\text{maxIter}} + a_1] \quad (21)$$

其中,  $a_1 = 0.5, a_2 = 2.5$ .

定义二元变量更新操作 DisOpe 如下: 将  $\boldsymbol{C}$  变换成  $N_{\text{nei}} \times N_{\text{sub}}$  的矩阵, 依次对比矩阵中每一列的值, 如果矩阵  $l$  列的值均相等, 则输出矩阵  $l$  列的值; 否则, 分别计算出所有矩阵每行第  $l$  个元素之和. 如果所有和中存在唯一最大值, 且位于第 row 行, 则把输出矩阵中第 row 行  $l$  列的值设为 1, 其他位设为 0; 否则, 从这些最值所在行中随机选择一行, 把输出矩阵中该行的第  $l$  列的值设为 1, 其他位设为 0. 根据 DisOpe ( $\boldsymbol{v}_{\text{CHL}}^{\text{old}}$ ) 和 DisOpe ( $\boldsymbol{v}_{\text{CLL}}^{\text{old}}$ ) 更新  $\boldsymbol{v}_{\text{HL}}^{\text{new}}$  和  $\boldsymbol{v}_{\text{LL}}^{\text{new}}$  的整数向量  $\boldsymbol{v}_{\text{CHL}}^{\text{new}}$  和  $\boldsymbol{v}_{\text{CLL}}^{\text{new}}$ .

$\boldsymbol{v}^{\text{new}}$  的实数向量  $\boldsymbol{v}_{\text{P}}^{\text{new}} = \boldsymbol{v}_{\text{PHL}}^{\text{new}} + \boldsymbol{v}_{\text{PLL}}^{\text{new}}$ , 整数向量  $\boldsymbol{v}_{\text{C}}^{\text{new}} = \text{DisOpe}(\boldsymbol{v}_{\text{C}}^{\text{old}})$ ,  $\boldsymbol{v}_{\text{C}}^{\text{old}}$  为探索燕子当前速度和整数部分.  $\boldsymbol{X}^{\text{new}}$  的实数向量和整数向量部分分别为  $\boldsymbol{X}_{\text{P}}^{\text{new}} = \boldsymbol{X}_{\text{P}}^{\text{old}} + \boldsymbol{v}_{\text{P}}^{\text{new}}$  和  $\boldsymbol{X}_{\text{C}}^{\text{new}} = \text{DisOpe}(\boldsymbol{X}_{\text{C}}^{\text{old}})$ ,  $\boldsymbol{X}_{\text{C}}^{\text{old}}$  为探索燕子当前位置的整数部分.

根据探索燕子  $\boldsymbol{X}^{\text{exp}}$  与局部领队燕子  $\boldsymbol{X}^{\text{LL}}$  之间的距离判断探索燕子是否属于某一局部领队燕子, 距离计算式如下:

$$\text{Dis}(\boldsymbol{X}^{\text{exp}}, \boldsymbol{X}^{\text{LL}}) = \sqrt{\sum_{i=1}^{N_{\text{nei}}} \sum_{j=1}^{N_{\text{sub}}} (c_{i,j}^{\text{exp}} - c_{i,j}^{\text{LL}})^2 + \sum_{i=1}^{N_{\text{sub}}} (\frac{p_i^{\text{exp}} - p_i^{\text{LL}}}{p_{\text{total}}})^2} \quad (22)$$

### 2.5 无目标燕子

把适应度函数值较差的燕子设置为无目标燕子, 采取随机飞行寻找被忽略的解空间. 飞行后的新位置  $\boldsymbol{X}^{\text{new'}}$  中实数向量更新方式如下:

$$\boldsymbol{X}_{\text{P}}^{\text{new'}} = \boldsymbol{X}_{\text{P}}^{\text{old'}} + [\text{rand}\{-1, 1\} \cdot \frac{\text{rand}(0, p_{\text{total}})}{1 + \text{rand}()}]. \quad (23)$$

其中:  $\text{rand}\{-1, 1\}$  表示随机生成  $-1$  或  $1$ ;  $\text{rand}(0, p_{\text{total}})$  表示生成  $[0, p_{\text{total}}]$  区间中的随机数.  $\boldsymbol{X}^{\text{new'}}$  的整数向量  $\boldsymbol{X}_{\text{C}}^{\text{new'}} = \text{DisOpe}(\boldsymbol{X}_{\text{C}}^{\text{old'}})$ , 其中  $\boldsymbol{X}^{\text{rand}}$  是按照解初始化方式随机生成的位置值.

### 2.6 算法伪代码

算法 1 低轨道星间带宽功率联合分配算法:

- 输入:  $p_{\text{total}}, N_{\text{sub}}, N_{\text{nei}}, N_{\text{lea}}, N_{\text{exp}}, N_{\text{aim}}, \text{maxIter}$
- 输出:  $\boldsymbol{C}$  和  $\boldsymbol{P}$
1. 初始化燕子群  $S_{\text{pop}}$  中燕子位置  $\boldsymbol{X}, \boldsymbol{v}_{\text{HL}}$  和  $\boldsymbol{v}_{\text{LL}}, \text{iter} = 1$ ;
  2. FOR  $\text{iter} \leq \text{maxIter}$ ;
  3. 构造动态可行域  $D_{\text{fea}}$ ;
  4. 按照式(16)计算所有个体的适应度函数

- 值  $\text{Fit}(\boldsymbol{X})$ ;
5. 根据  $\text{Fit}(\boldsymbol{X})$  分类种群  $S_{\text{pop}}$  中的燕子;
  6. 对于探索燕子根据式(22)找出距离最近的局部领队燕子;
  7. 计算  $\alpha_{\text{HL}}, \beta_{\text{HL}}, \alpha_{\text{LL}}$  和  $\beta_{\text{LL}}$ , 并更新探索燕子位置;
  8. 更新无目标燕子位置;
  9.  $\text{iter}++$ ;
  10. END FOR
  11. 根据最优个体计算  $\boldsymbol{C}$  和  $\boldsymbol{P}$ , 并返回.

## 3 仿真与性能分析

采用 STK 为仿真提供卫星运动轨迹、可见性和拓扑数据. 用仿真平台 Eclipse 实现最大化卫星瞬时吞吐容量的资源分配方法 (maximum capacity based resources allocation method, MCRAM) 和基于比例公平性的资源分配方法 (proportional fairness based resources allocation method, PFRAM). 铱星系统为极轨道星座, 全球星系统为 Walker 星座, 故本文采用上述两种星座作为仿真拓扑用例. 仿真参数如表 1 所示.

表 1 仿真参数

Table 1 Simulation parameters

参数名称	参数值 (铱星/ 全球星)	参数名称	参数值 (铱星/ 全球星)
卫星数	66/48	轨道面内卫星数	11/6
轨道面数	6/8	轨道高度/km	780/1 414
相位因子	—/2	系统周期/min	100.45/114.09
轨道倾角/(°)	86.4/52	边界纬度值/(°)	60/48
带宽/Mbps	200/200	领队燕子数	20
探索燕子数	250	无目标燕子数	50
地面站数	100	最大迭代次数	1 000

### 3.1 卫星瞬时吞吐容量

从铱星系统和全球星系统周期中分别均匀选取 154 和 144 个采样点, 并随机选取 1 颗从高纬地区向低纬地区运动的卫星, 分析其瞬时吞吐容量变化, 结果如图 1 所示. 由图 1 可知, 铱星系统卫星瞬时吞吐容量曲线开始平稳, 因为卫星位于高纬地区, 只有 2 条距离不变的同轨链路. 当卫星向赤道运行时, 两异轨链路重新建立, 卫星瞬时吞吐容量变大. 当卫星靠近赤道时, 异轨链路距离变大, 卫星瞬时吞吐容量减小. 卫星经过赤道后, 异



轨链路距离变小,卫星瞬时吞吐容量变大.再次进入高纬度地区时,异轨链路断开,卫星瞬时吞吐容量迅速减小,上述过程周期性重复出现. PFRAM 考虑了地面站的影响,曲线在整个周期内出现随机波动.全球星系统的卫星瞬时吞吐容量曲线波动较复杂,因为其异轨链路断开或重连的时刻不同,当一端的卫星进入高纬度地区时,卫星仍能保持 3 条通信链路,导致曲线出现先下降后升高的情况.

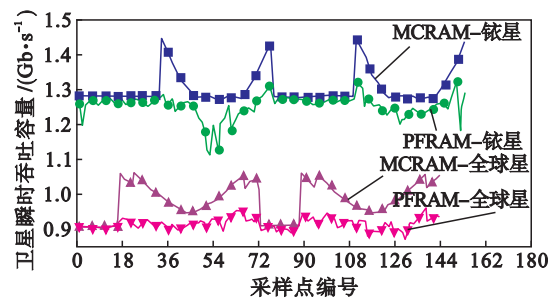


图 1 卫星瞬时吞吐容量  
Fig. 1 Satellite instantaneous throughput capacity

3.2 卫星平均吞吐容量对比

卫星功率为 20,60,100 W 时,两卫星系统的卫星平均吞吐容量如图 2 所示.由图 2 可知,增加卫星功率可提高卫星平均吞吐容量.当卫星功率相同时,MCRAM 的卫星平均吞吐容量较大,是因为 PFRAM 考虑了卫星链路间的公平性,增强了约束性,导致卫星平均吞吐容量偏小,但在整个周期内卫星覆盖的地面站数和链路可用时间不断变化,导致两种方法的卫星平均吞吐容量差异较小. PFRAM 充分考虑覆盖的地面站数和星地链路存在时间,均衡合理地分配更多资源给覆盖区域内业务需求多且链路稳定的卫星.因此,PFRAM 的卫星平均吞吐容量虽然降低了,但提高了资源分配的公平性及合理性.

3.3 瞬时网络容量对比

分析两卫星系统周期内的瞬时网络容量,仿真结果如图 3 所示.由图 3 可知,MCRAM 的瞬时网络容量曲线具有很强的规律性,但由于地面站分布不均,PFRAM 的曲线伴有一定的随机波动.这是因为系统中星际链路频繁切换,导致瞬时网络容量增大或减小.当有较多卫星在高纬度地区时,星际链路较少,星上资源得不到有效利用,瞬时网络容量较小.当这些卫星离开高纬度地区时,链路重连,瞬时网络容量增大.随着卫星不断进出高纬度地区,瞬时网络容量值规律性波动,但由于链路长度变化导致轻微波动.

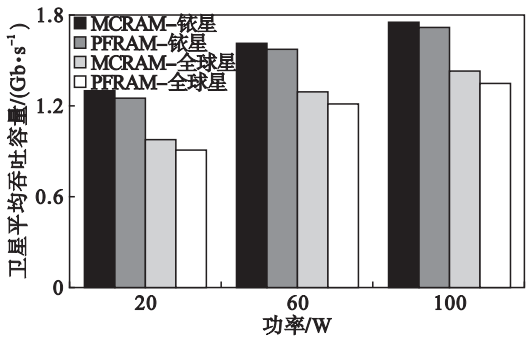


图 2 平均吞吐容量  
Fig. 2 Average throughput capacity

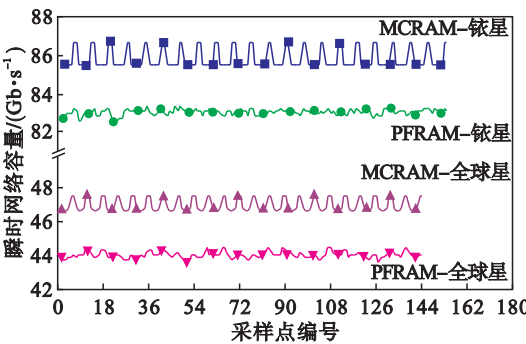


图 3 瞬时网络容量  
Fig. 3 Instantaneous network capacity

3.4 平均网络容量对比

卫星功率为 20,60,100 W 时,两卫星系统的平均网络容量如图 4 所示.由图 4 可知,增加卫星功率可提高平均网络容量.当卫星功率相同时,MCRAM 的平均网络容量较大,这是因为 PFRAM 考虑卫星链路间的公平性,增强了约束性,导致平均网络容量偏小.资源分配的公平性保证了地面站数较多、星地链路存在时间较长的区域能获得更多的网络资源,服务质量提高;反之分配较少的网络资源,但在整个周期内卫星覆盖的地面站数和链路可用时间不断变化,导致两种方法的平均网络容量差异较小. PFRAM 通过牺牲平均网络容量获得网络资源分配公平性,提高了资源利用率.

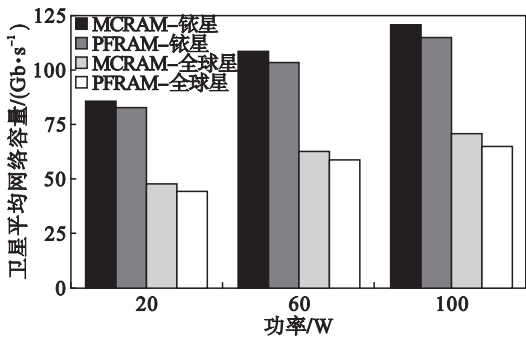


图 4 平均网络容量  
Fig. 4 Average network capacity