中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION

面向低轨星座馈电链路切换问题的混合克隆选择算法

任思达, 冯彦翔, 陈炜, 张广辉, 杨宜康

引用本文:
任思达,冯彦翔,陈炜,张广辉,杨宜康. 面向低轨星座馈电链路切换问题的混合克隆选择算法[J]. 控制与决策, 2024, 39(10): 3385−3394.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2023.0663

面向低轨星座馈电链路切换问题的混合克隆选择算法

任思达[1], 冯彦翔[1†], 陈 炜[2], 张广辉[3], 杨宜康[1]

(1. 西安交通大学 自动化科学与工程学院，西安 710049；2. 河北农业大学 人事处，河北 保定 071001；
3. 河北农业大学 信息科学与技术学院，河北 保定 071001)

摘 要: 随着低轨星座规模的不断扩张,本就相对匮乏的信关站资源变得更加紧张. 为提高信关站天线的使用效率,提出一种混合克隆选择算法. 首先,将卫星与信关站可见弧段转化为任务集合,将馈电链路切换问题转化为任务分配问题,并建立相应的数学整数规划模型;然后,将抗体编码为一组任务分配向量,结合启发式冲突消解规则,建立基于有向图最短路的解码方法,引入阈值参数降低解码的计算开销,提出基于自适应邻域选择的局部搜索算法,增强局部寻优能力;最后,搭建低轨星座馈电链路切换仿真场景,生成不同规模的算例来开展对比实验. 仿真实验结果表明,所提出算法能够快速收敛到小规模算例的最优解,同时在大规模算例上比现有启发式算法表现出更强的求解能力和更稳定的性能,从而验证所提出算法的有效性.

关键词: 低轨星座；馈电链路切换；克隆选择算法；有向图；任务分配；元启发式算法

中图分类号: TP273 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2023.0663

引用格式: 任思达,冯彦翔,陈炜,等. 面向低轨星座馈电链路切换问题的混合克隆选择算法[J]. 控制与决策, 2024, 39(10): 3385-3394.

Hybrid clonal selection algorithm for feeder link handover problem in LEO satellite constellation

REN Si-da[1], *FENG Yan-xiang*[1†], *CHEN Wei*[2], *ZHANG Guang-hui*[3], *YANG Yi-kang*[1]

(1. School of Automation Science and Engineering，Xi'an Jiaotong University，Xi'an 710049，China；2. Human Resources Office，Hebei Agricultural University，Baoding 071001，China；3. School of Information Science and Technology, Hebei Agricultural University，Baoding 071001，China)

Abstract: The continuous expansion of LEO satellite constellation has made the relatively scarce gateway stations even more strained. To improve the utilization rate of gateway station antennas, a hybrid clonal selection algorithm is proposed. First, by transforming visible arcs into a task set, the feeder link handover problem is transformed into a task assignment problem, and the corresponding integer programming model is established. Then the antibody is encoded as a set of task assignment vectors, and a decoding method based on the shortest path in directed graphs is proposed in combination with heuristic conflict resolution rules. A threshold parameter is introduced to reduce the computation cost of the decoding method. Furthermore, an adaptive neighborhood selection-based local search algorithm is proposed to enhance local optimization capabilities. Finally, the simulation scenario of feeder link handover in LEO satellite constellation is constructed, and instances of different scales are generated. Experimental results show that the proposed algorithm can quickly converge to the optimal solution for small-scale instances, while showing stronger solving ability and more stable performance than the existing heuristics on large-scale instances, thus verifying its effectiveness.

Keywords: LEO satellite constellation；feeder link handover；clonal selection algorithm；directed graphs；task assignment；meta-heuristic algorithm

0 引 言

兴起于20世纪末的低轨(卫星)星座凭借广覆盖、全天候的优势,成为未来空天地一体化网络的重要组成部分[1]. 近年来,随着卫星批量化制造以及发射成本的降低,国内外掀起了新一轮低轨星座网络的发展热潮[2].

收稿日期: 2023-05-16；**录用日期:** 2023-10-16.

基金项目: 2020年度科技创新2030—“新一代人工智能”重大项目(2020AAA0108200).

责任编委: 刘民.

†通讯作者. E-mail: fengyanxiang@xjtu.edu.cn.

*本文附带电子附录文件,可登录本刊官网该文“资源附件”区自行下载阅览.

作为低轨星座系统地面段的主体之一,信关站只能在某些特定的时间段(或弧段)通过天线与卫星建立馈电链路. 为了实现地面网与卫星网的持续连接,信关站需要在不同的可见卫星之间切换馈电链路. 受限于建设成本和国际政治等因素,目前我国信关站数量较少,且均位于国境内. 随着新兴低轨星座规模的不断扩张,其链路需求使得本就相对匮乏的信关站资源变得更加紧张. 所以,如何高效实现信关站天线与卫星的馈电链路连接与切换,已成为低轨星座系统的重要问题.

低轨星座馈电链路切换问题本质上属于SRSP (satellite range scheduling problem),已被证明是NP完全的[3]. SRSP以任务收益最大为调度目标,在可见弧段内连接卫星与信关站的天线,从而完成非抢占性通信任务. SRSP求解方法包括精确算法和近似算法两类,其中精确算法在小规模问题上表现出色,但在大规模问题上受限于海量的参数和约束条件,无法在可接受时间内获得结果[4]. 因此,研究者往往采用启发式[5-6]和元启发式[7-8]等方法处理大规模SRSP. 比如:文献[7]利用图来表示可见弧段的冲突关系,处理了任务执行唯一性和可见弧段无重叠等约束;文献[9]基于爬山法框架,实现禁忌搜索、模拟退火和逾期接受算法的混合;文献[8,10]考虑了包括任务冲突度、信关站使用效率等目标,分别利用模拟退火和遗传算法进行求解;文献[11]将可见弧段定义为任务,利用蚁群算法实现任务执行时间的最大化;文献[12]将遗传算法与粒子群算法相结合,利用交叉和变异算子更新粒子,并引入虚拟卫星来增强搜索能力;文献[13]在人工蜂群算法中引入了学习机制,高效利用优化过程获得的有用信息.

SRSP中多个任务会争夺同一信关站(天线)资源,出现冲突. 现有文献[4-13]在处理任务冲突时,根据启发式规则选择一个任务而放弃其他任务,这样明显不能充分利用信关站资源,存在资源浪费的问题. 为了更灵活地处理冲突、提高资源使用效率,一种可行方式是“分割”任务,将“未执行”的任务再次分配给信关站天线. 但这会进一步增加问题的求解难度,从而对调度算法的设计提出了新的挑战.

相较于其他启发式和元启发式算法,克隆选择算法(clonal selection algorithm, CSA)凭借适用范围广、收敛速度快、全局搜索能力强等优势,广泛应用于成像卫星调度问题[14]、卫星测控资源调度问题[15]等. 受限于自身结构,基本CSA在求解复杂问题时可能陷入局部最优. 针对这一问题,一种方法是改进CSA的算子设计,如文献[16]提出一种随进化代数增长自适应调整领域范围的变异算子. 但在大规模问题上,这种方法效果有限. 将CSA与其他算法混合,利用算法的优势互补来提升搜索能力,是另一种有效的途径. 文献[17]将CSA与变邻域搜索相结合,有效提升了算法的局部搜索能力.

低轨星座馈电链路切换问题规模大、复杂度高,基于任务分割处理冲突进一步增加了求解难度. 受CSA在求解相关问题上的出色表现启发,本文提出一种混合克隆选择算法,将CSA与局部搜索算法结合,高效解决馈电链路切换问题. 首先,将卫星-信关站可见弧段转化为彼此独立的任务,成功将链路切换问题转化为天线任务分配问题,并建立了相应数学整数规划模型; 然后,设计了CSA的编码、解码和算子,并提出一种自适应邻域选择的局部搜索算法; 最后,利用STK软件搭建仿真实验场景. 实验结果表明,所提出算法能够快速得到小规模算例的最优解,同时在大规模算例上体现出比现有算法[11,13,18-20]更强的求解能力和更稳定的性能. 本文主要工作如下:

1) 将卫星与信关站的可见弧段转化为相互独立的任务,并设计了启发式规则用于处理任务冲突.

2) 提出一种基于有向图最短路的抗体解码方法,通过引入阈值参数降低了这种解码法的计算开销.

3) 引入一种自适应邻域选择的局部搜索策略,显著增强了算法的局部寻优能力.

1 问题描述

假设低轨星座系统共有m颗卫星$S=\{1,2,\dots,m\}$和n个地面信关站$U=\{1,2,\dots,n\}$,每个信关站有a副天线可与卫星建立馈电链路,天线集合是$K=\{1,2,\dots,n\times a\}$. 为方便起见,本文用$s\in S,u\in U$和$k\in K$分别表示一颗卫星、信关站和天线.

信关站u中的天线集合用$\Delta(u)=\{(u-1)a+1,(u-1)a+2,\dots,(u-1)a+a\}$来表示. 在调度周期$[0,T]$内,卫星与信关站的可见性用$v_{s,u,t}$来描述,如果时隙$[t-1,t]$内卫星$s$与信关站$u$内所有天线可见,则令$v_{s,u,t}=1$,反之则令$v_{s,u,t}=0$. 多个连续的可见时隙组成了信关站与卫星的“可见弧段”. 在可见时隙内,卫星可与信关站的任意一副空闲天线建立馈电链路. 本文的目标是在满足卫星-信关站可见性约束条件下,尽可能地延长卫星与信关站天线的连接时间.

1.1 任务构建

本文提出算法1,将卫星-信关站的可见性变量$v_{s,u,t}(\forall s\in S,u\in U,t\in T)$转化为“任务”集合$G$,

通过将G中的任务分配给天线,求解低轨星座馈电链路切换问题. 每个天线任务$g \in G$可用三元组$\langle s_g, Q_g, [t_g, e_g]\rangle$表示,其中$s_g$是与$g$关联的唯一卫星,$Q_g \subseteq K$是可执行任务$g$的天线集合,$[t_g, e_g]$是可见时间窗. 执行任务$g$等同于在$[t_g, e_g]$内,$Q_g$中的天线与卫星$s_g$建立馈电链路. 具体见算法1.

算法1　TaskGeneration.

输入: 可见性变量$v_{s,u,t}$;
输出: 任务集合G.
1) 令$v_s(s,t)=\{u\in U|v_{s,u,t}=1\}$;
2) 令$G=\varnothing, g=1$;
3) foreach $s\in S$ do
4)　for $0\leqslant t<T$ do
5)　if $v_s(s,t)\neq v_s(s,t+1)$ then
6)　if $v_s(s,t)\neq\varnothing$ then
7)　$e_g=t; s_g=s$;
8)　$Q_g=\{\Delta(u)|u\in v_s(s,t)\}$
9)　$g=g+1$;
10)　end
11)　if $v_s(s,t+1)\neq\varnothing$ then
12)　$G:=G\bigcup\{g\}, t_g=t$
13)　end
14)　end
15)　end
16)　if $v_s(s,T)\neq\varnothing$ then
17)　$e_g=T, s_g=s$;
18)　$Q_g=\{\Delta(u)|u\in v_s(s,T)\}, g=g+1$;
19)　end
20) end
21) 输出任务集合G.

在算法1中,$v_s(s,t)=\{u\in U|v_{s,u,t}=1\}$表示$[t-1,t]$时隙内卫星$s$的可见信关站集合,为方便起见,初始时令$v_s(s,0)=\varnothing$. 算法1根据卫星运行过程中可见信关站的变化情况,将卫星-信关站可见弧段划分为不同任务. 输出的任务集合G中关联相同卫星的任务可见时间窗不会重合,即对于任务g_1和g_2,如果$s_{g_1}=s_{g_2}$,则$[t_{g_1},e_{g_1}]\bigcap[t_{g_2},e_{g_2}]=\varnothing$.

1.2　数学模型

本文将星座馈电链路切换问题转化为天线任务分配问题,即把G中的任务分配给K中的天线. 注意同一个任务可能被分配给多副天线.

令$\Pi_k=\langle g_1,g_2,\ldots,g_{|\Pi_k|}\rangle$表示分配给天线$k\in K$的任务序列,假设$k$会按照顺序依次执行$\Pi_k$中的任务,令$T_g^k$和$E_g^k$表示任务$g\in\Pi_k$被$k$执行的开始时间和结束时间. 所有天线任务分配序列构成了一个解$\boldsymbol{\Pi}=[\Pi_1,\ldots,\Pi_{n\times a}]$,需要满足以下约束条件:

1) 任务g只能被Q_g中的天线执行,有

$$k\in Q_g,\ \forall k,\forall g\in\Pi_k. \tag{1}$$

2) 任务g只能在可见时间窗内被执行,有

$$t_g\leqslant T_g^k\leqslant E_g^k\leqslant e_g,\ \forall k,\forall g\in\Pi_k. \tag{2}$$

3) 天线转向时间约束,天线执行完成一个任务后,若下一个任务的关联卫星与当前任务不一样,则需要时间HT完成方向调整,即

$$T_{g_{x+1}}^k-E_{g_x}^k\geqslant\Xi(g_x,g_{x+1}),\quad \forall k,\forall x\in\{1,2,\ldots,|\Pi_k|-1\}. \tag{3}$$

其中:$\Xi(g_x,g_{x+1})$表示连续任务g_x与g_{x+1}之间的调整时间,有

$$\Xi(g_x,g_{x+1})=\begin{cases}\text{HT},\ s_{g_x}\neq s_{g_{x+1}};\\ 0,\ \text{otherwise.}\end{cases} \tag{4}$$

式(2)和(3)保证天线k在某时刻最多执行一个任务.

4) 天线执行一个任务的时间不能小于最小连接时间PT,即

$$E_g^k-T_g^k\geqslant\text{PT},\ \forall k,\forall g\in\Pi_k. \tag{5}$$

5) 任务g某时刻最多被一副天线执行,即

$$[E_g^k-T_g^l][E_g^l-T_g^k]\leqslant 0,\quad \forall k,l\in K\text{ with }k\neq l,\ \forall g\in\Pi_k\bigcap\Pi_l. \tag{6}$$

本文所研究问题的数学规划模型如下:

$$\max\ J=\sum_{k\in K}\sum_{g\in\Pi_k}(E_g^k-T_g^k);\quad \text{s.t. 式}(1)\sim(6). \tag{7}$$

其中:优化目标(7)极大化所有任务的执行时间,即在满足所有约束的前提下,在可见弧段内尽可能地延长卫星与天线的连接时间.

1.3　执行时间窗计算

如图1(a)所示,天线k先执行任务$g\in\Pi_k$再执行任务$h\in\Pi_k$时,可能会发生时间窗冲突,即任务g和h对于天线k可见时间窗有重合. 现有文献[4-13]在处理此类情况时,都是选择任务g或h,而舍弃另一个任务. 这明显会延长天线空闲时间,造成资源浪费. 针对这个问题,本文提出一种冲突消解算法2,在满足约束(2)~(5)的前提下,确定任务g的结束时间E_g^k,通过重新计算h的可见时间窗$[t_h,e_h]$,分割任务h. 分割后的任务h还能分配给其他天线,尽可能地利用资源.

在算法2中,受文献[18]启发,首先令$E_g^k=e_g$,然

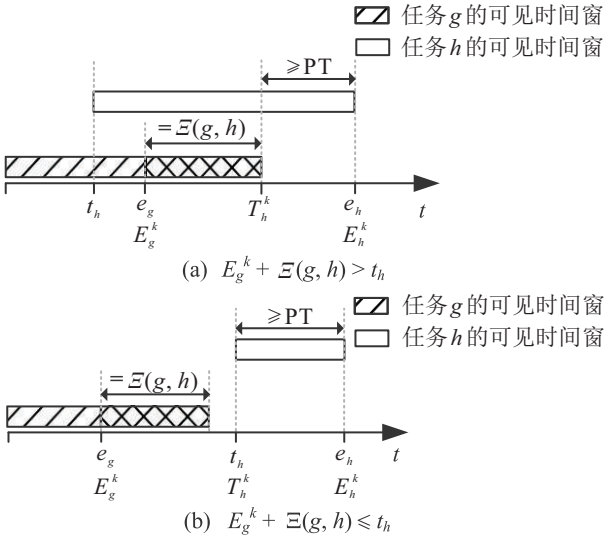

图 1　任务$g,h\in\Pi_k$的可见时间窗分布

算法2　ConflictResolution.

输入: Π_k中的连续两个任务g和h;
输出: E_g^k、T_h^k和E_h^k.
1) 令$E_g^k=e_g$;
2) if $e_h-\max\{E_g^k+\Xi(g,h),t_h\}\geqslant$ PT then
3)　　令$T_h^k=\max\{E_g^k+\Xi(g,h),t_h\}$;
4)　　令$E_h^k=e_h$;
5) else
6)　　令$T_h^k=E_h^k=0$;
7) end
8) 输出E_g^k、T_h^k和E_h^k.

后根据下式是否成立,分两种情况讨论:

$$e_h-\max\{E_g^k+\Xi(g,h),t_h\}\geqslant \text{PT}. \tag{8}$$

Case 1: 式(8)成立. 当$E_g^k+\Xi(g,h)>t_h$时(如图1(a)所示),令$T_h^k=E_g^k+\Xi(g,h)$, $E_h^k=e_h$,由式(8)可知满足约束(5);若$E_g^k+\Xi(g,h)\leqslant t_h$(如图1(b)所示),则令$T_h^k=t_h$, $E_h^k=e_h$,同理可知约束(5)满足.

Case 2: 式(8)不成立. 令$T_h^k=E_h^k=0$,表示任务h不存在满足约束的执行时间窗,可认为天线k无法执行该任务.

2　混合克隆选择算法

本文提出一种混合克隆选择算法(hybrid clonal selection algorithm, HCSA),用于解决低轨星座馈电链路切换问题. 经典的克隆选择算法CSA[21]将优化问题及可行解分别等效为免疫系统中的抗原与抗体,将寻优过程表征为抗体进化过程,具有较强的全局搜索能力.

2.1　抗体编码和解码

1) 编码. 假设种群P含有N_p个抗体. 每个抗体$\pi=\{\pi_1,\pi_2,\dots,\pi_{n\times a}\}$包含$n\times a$个子抗体,子抗体$\pi_k$是$|W_k|$维二进制向量,表示任务分配给天线$k$的情况,其中$W_k=\{g\in G|k\in Q_g\}$是天线$k$可执行的任务集合. 对于任务$g\in W_k$, π_k的分量$\pi_k[g]=1$表示将任务g分配给天线k,否则$\pi_k[g]=0$. 用$w_k=\{g\in W_k|\pi_k[g]=1\}$表示子抗体$\pi_k$对应的分配给天线$k$的任务集合.

例 1　将6个任务分配给4副天线,调度周期为[0,12]. 表1给出了任务属性,可知$W_1=W_2=\{1,2,3,5\}$, $W_3=W_4=\{1,3,4,5,6\}$. 图2给出了一个抗体π的编码,可知$w_1=\{1,2,3,5\}$, $w_2=\{1,2,3\}$, $w_3=\{1,3\}$, $w_4=\{1,4,5\}$.

表 1　任务属性

任务g	s_g	Q_g	$[t_g,e_g]$
1	1	{1,2,3,4}	[0,8]
2	1	{1,2}	[8,12]
3	2	{1,2,3,4}	[0,9]
4	2	{3,4}	[9,12]
5	3	{1,2,3,4}	[0,6]
6	3	{3,4}	[6,12]

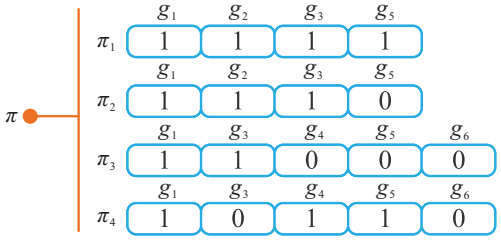

图 2　抗体π的编码

2) 解码. w_k表示从子抗体π_k中得到的分给天线k的任务集合,没有执行顺序. 本文基于有向图最短路径思想,进一步从w_k解码得到天线k需要执行的任务序列Π_k,具体步骤如下.

首先,利用算法3构造有向图$\text{Graph}_k=(V,A,f)$,节点集合$V=\{v_0,v_1,\dots,v_{|w_k|},v_{|w_k|+1}\}$,其中$\{v_1,\dots,v_{|w_k|}\}$与$w_k$中的任务一一对应, v_0和$v_{|w_k|+1}$为源

算法3　GraphConstruct.

输入: 任务集合w_k;
输出: 有向图$\text{Graph}_k=(V,A,f)$.
1) 令$V=\{v_0,v_1,\dots,v_{|w_k|},v_{|w_k|+1}\}$, $A=\varnothing$
2) forEach $v_g,v_h\in V$ with $g\neq h$ do
3)　　根据算法2计算T_h^k、E_h^k和E_g^k;
4)　　if $E_h^k>T_h^k$且$T_h^k-E_g^k\leqslant$ WT then
5)　　$A:=A\bigcup\langle v_g,v_h\rangle$;
6)　　$f(v_g,v_h)=T_h^k-E_g^k$;
7)　　end
8) end
9) 输出Graph_k.

节点和目的节点, 分别对应虚拟开始任务 g_s 和结束任务 g_e, 规定它们的可见时间窗分别是 $[-\mathrm{PT}, 0]$ 和 $[T, T+\mathrm{PT}]$. 此外, 令 g_s 和 g_e 与其他任务之间的调整时间为0.

在算法3中, $\forall v_g, v_h \in V$ 且 $g \neq h$, 调用算法2, 若能得到任务 h 的有效执行时间窗 $[T_h^k, E_h^k]$, 则从 v_g 到 v_h 可添加一条弧线 $\langle v_g, v_h\rangle$, 其权重为 $f(v_g, v_h) = T_h^k - E_g^k$, 表示天线 k 在执行完 g 后, 到执行 h 之前处于空闲状态的时间. 为了降低最短路径算法的开销, 引入了阈值参数WT, 后续实验第3.5节验证了WT的有效性.

然后, 在有向图 Graph_k 中, 弧线 $\langle v_g, v_h\rangle$ 表示天线 k 先执行 g 再执行 h. 根据经典 Dijkstra 算法, 在 Graph_k 上寻找一条从 v_0 到 $v_{|w_k|+1}$ 的权重之和最小的路径 $p = v_0 \to v_a \to v_b \ldots \to v_c \to v_{|w_k|+1}$, 根据 p 得到相应的任务序列 $\Pi_k = \langle g_a, g_b, \ldots, g_c\rangle$. 所有天线解码出的任务序列构成一个解 $\boldsymbol{\Pi} = [\Pi_1, \ldots, \Pi_{n\times a}]$.

例2　令 $\mathrm{HT}=3, \mathrm{PT}=1, \mathrm{WT}=6$, 对图2中的子抗体 π_1 进行解码. 调用算法3构造的有向图 Graph_1 如图3所示, 其中节点 v_0 和 v_5 分别是源节点和目的节点, $v_1 \sim v_4$ 对应 w_1 中的任务1、2、3和5. 利用Dijkstra算法找到从 v_0 到 v_5 的最短路径为 $v_0 \to v_1 \to v_2 \to v_5$, 对应任务序列 $\Pi_1 = \langle 1, 2\rangle$, 表示天线1顺序执行任务1和任务2.

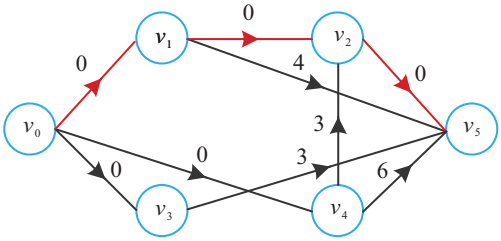

图 3　任务集合 w_1 对应的有向图 Graph_1

2.2　抗体亲和度计算

抗体-抗原亲和度(简称为抗体亲和度)用于评价解的质量. 抗体 π 的亲和度 $F(\pi)$ 按照下式计算:

$$F(\pi) = \frac{\sum\limits_{k\in K}\sum\limits_{g\in \Pi_k}(E_g^k - T_g^k)}{\sum\limits_{g\in G}(e_g - t_g)}. \tag{9}$$

亲和度 $F(\pi)$ 越高, 表示所有卫星与信关站的连接时间越长, 则 π 的质量越好. 根据2.1节的分析, 给出算法4计算每个抗体 π 的亲和度 $F(\pi)$.

算法4首先从抗体 π 解码出天线 k 的任务序列 Π_k, 然后依次为 Π_k 中的任务计算执行时间窗, 并重置其可见时间窗以满足约束(6). 最后, 利用式(9)计算并输出抗体 π 的亲和度 $F(\pi)$.

算法4　AffinityCalculation.

输入: 抗体 π;
输出: 抗体亲和度 $F(\pi)$.
1) 从抗体 π 编码得到任务分配集合 $[w_1, \ldots, w_{n\times a}]$;
2) for $1 \leqslant k \leqslant n \times a$ do
3)　对于 w_k, 调用算法3构造有向图 Graph_k, 其中 $\mathrm{WT} = 20 \times \mathrm{HT}$;
4)　在 $\mathrm{Graph_k}$ 调用Dijkstra得到 $\Pi_k = \{g_1, g_2, ..., g_{|\Pi_k|}\}$;
5)　foreach $g_x \in \Pi_k$
6)　　if $x = 1$ then
7)　　$T_{g_x}^k = t_{g_x}$;
8)　　else
9)　　$T_{g_x}^k = \max\{E_{g_{x-1}}^k + \Xi(g_{x-1}, g_x), t_{g_x}\}$;
10)　　end
11)　　$E_{g_x}^k = e_{g_x}$;
12)　　$e_{g_x} = T_{g_x}^k$; //重置可见时间窗
13)　end
14) end
15) 利用式(9)计算 $F(\pi)$;
16) 输出 $F(\pi)$.

2.3　基于自适应邻域选择的局部搜索

为增强算法的局部寻优能力, 本文提出一种基于自适应邻域选择的局部搜索算法, 搜索所有抗体的3种邻域结构. 首先介绍3种邻域动作和对应的邻域结构.

1) add动作: 随机选择抗体 π 中编码值为0的分量 $\pi_k[g]$, 将其翻转为1. 用 $N_1(\pi)$ 表示抗体 π 的add动作邻域结构.

2) remove动作: 随机选择任务 $g \in \Pi_k$, 将其对应的编码 $\pi_k[g]$ 翻转为0, 对应的邻域结构为 $N_2(\pi)$.

3) exchange动作: 随机选择两根天线 $k, l \in Q_g$, 满足 $\pi_k[g] \neq \pi_l[g]$, 交换 $\pi_k[g]$ 和 $\pi_l[g]$ 的值, 表示将任务 g 进行重分配, 对应的邻域结构为 $N_3(\pi)$.

在每轮迭代中, 局部搜索算法将随机选择一种邻域结构进行搜索, 用 $p_r(r \in \{1,2,3\})$ 表示邻域结构 N_r 被选择的概率. 为充分利用迭代过程中的有用信息, 本文提出一种自适应邻域选择策略, 在迭代过程中动态更新邻域结构的选择概率. 具体来说, 每经过25轮迭代后, $p_r(r \in \{1,2,3\})$ 将按照下式进行更新:

$$p_r' = \zeta p_r + (1-\zeta)\frac{\mathrm{suc}_r}{\mathrm{sel}_r}, \tag{10}$$

$$p_r = \frac{p_r'}{\sum\limits_{m=1}^{3} p_m'}. \tag{11}$$

其中: $\zeta = 0.7$ 为惯性因子; sel_r 为邻域结构 N_r 在前

25轮迭代中被选择的次数,suc_r是在前25轮迭代中利用N_r生成更优抗体的次数.

基于以上分析,给出描述局部搜索算法的算法5,其中ψ为当前迭代次数. 每轮迭代首先利用轮盘赌选择一种邻域结构,从中随机选择抗体π',若$F(\pi') > F(\pi_{\text{best}})$,则将$\pi_{\text{best}}$更新为$\pi'$. 每经过25轮迭代,更新$p_r$、$\text{sel}_r$和$\text{suc}_r, r \in \{1,2,3\}$. 最后,当$\psi$大于阈值$\Psi$时,算法结束并输出$\pi_{\text{best}}$.

算法5 LocalSearch.

输入: 抗体$\pi, p_r, r \in \{1,2,3\}$,阈值$\Psi$;
输出: 局部最优抗体π_{best}.
1) 令$\psi = 1, \pi_{\text{best}} = \pi, p_1 = p_2 = p_3 = 1/3$;
2) 令$\text{sel}_r = \text{suc}_r = 0, r \in \{1,2,3\}$;
3) while $\psi \leqslant \Psi$ do
4) if $D[\psi/25] = 0$ then
5) // $D[\cdot]$为取模,比如$D[6/4] = 2$
6) 按照式(10)和(11)更新$p_r, r \in \{1,2,3\}$;
7) 令$\text{sel}_r = \text{suc}_r = 0, r \in \{1,2,3\}$;
8) end
9) 利用轮盘赌随机选择一种邻域结构N_r;
10) $\text{sel}_r = \text{sel}_r + 1$;
11) 随机选择抗体$\pi' \in N_r(\pi_{\text{best}})$;
12) if $F(\pi') > F(\pi_{\text{best}})$ then
13) $\pi_{\text{best}} = \pi'$, $\text{suc}_r = \text{suc}_r + 1$;
14) end
15) $\psi = \psi + 1$;
16) end
17) 输出抗体π_{best}.

2.4 克隆选择算子

2.4.1 抗体激励度

激励度是评估抗体质量的综合指标,需要同时考虑抗体的亲和度和浓度.

首先计算种群P中两个抗体π和π'的亲和度$\text{aff}(\pi,\pi')$,以表示这两抗体的相似程度,即

$$\text{aff}(\pi,\pi') = \sum_{k\in K}\sum_{g\in W_k}\partial_k(\pi,\pi',g). \tag{12}$$

其中

$$\partial_k(\pi,\pi',g) = \begin{cases} 1, & \pi_k[g] = \pi'_k[g]; \\ 0, & \text{otherwise}. \end{cases} \tag{13}$$

抗体π与π'的相似度为

$$R(\pi,\pi') = \begin{cases} 1, & \text{aff}(\pi,\pi') \Big/ \sum\limits_{k=1}^{n\times a}|W_k| > \delta; \\ 0, & \text{otherwise}. \end{cases} \tag{14}$$

其中$\delta = 0.5$为预先设定的相似度阈值.

抗体π在种群中的浓度为

$$\text{den}(\pi) = \frac{1}{N_p}\sum_{\pi'\in P} R(\pi,\pi'). \tag{15}$$

$\text{den}(\pi)$过高意味着种群内存在大量与π相似的抗体,不利于全局寻优. 因此,需要抑制浓度过高的抗体,给予亲和度大且浓度低的抗体更高的激励度. 抗体π的激励度为

$$\text{sim}(\pi) = \text{AF}(\pi) \times \text{e}^{-\text{den}(\pi)/\beta}. \tag{16}$$

其中: $\beta = 2$为激励度调节因子,$\text{AF}(\pi)$如下计算:

$$\text{AF}(\pi) = \frac{F(\pi) - F_{\min}}{F_{\max} - F_{\min}}, \tag{17}$$

$F_{\max}$和$F_{\min}$分别是种群P中抗体亲和度的最大值和最小值.

2.4.2 克隆算子

将抗体π克隆复制$\text{cls}(\pi)$份,其中$\text{cls}(\pi)$称为π的克隆规模,有

$$\text{cls}(\pi) = M\Big(N_c\frac{\text{sim}(\pi)}{\sum\limits_{\pi'\in P}\text{sim}(\pi')} + \varphi\Big). \tag{18}$$

其中: $N_c = 2 \times N_p$为种群克隆规模; $\varphi = 1.2$是常数,保证每个抗体都有一定的克隆数量; $M[\cdot]$为向下取整函数,比如$M[5.6] = 5$.

利用克隆算子将P中每个抗体$\pi \in P$克隆复制$\text{cls}(\pi)$份,构成π的克隆抗体群$\text{clone}(\pi)$. 所有克隆抗体构成克隆种群P_c,实现原始种群的扩张.

2.4.3 变异算子

对抗体进行变异操作,以产生亲和度突变,实现局部搜索. 克隆抗体$\pi \in P_c$的变异概率$p_m(\pi)$为

$$p_m(\pi) = \Big[\exp\Big(N_p\frac{\text{AF}(\pi)}{\sum\limits_{\pi'\in P}\text{AF}(\pi')}\Big) + \rho\Big]^{-1}, \tag{19}$$

其中$\rho = 1$为变异概率调节因子. 式(19)表明,变异概率$p_m(\pi)$取决于抗体亲和度$F(\pi)$的大小,亲和度越大的抗体变异概率越小.

对于抗体π的每一位编码$\pi_k[g]$,变异算子将随机生成一个区间[0,1]上的均匀分布随机数rand,若$\text{rand} < p_m(\pi)$,则将$\pi_k[g]$的值翻转,即令$\pi_k[g] = 1 - \pi_k[g]$. 对克隆种群P_c中所有抗体调用变异算子,生成变异种群P'_c.

2.4.4 克隆抑制和种群刷新算子

将变异种群P'_c和原始种群P一起进行抗体亲和度评估后,筛选出优质抗体,并与随机生成的新抗体一起构成下一代种群P'. 该算子包括两个步骤: 克隆抑制和种群刷新.

1) 克隆抑制. 首先令$P' = P$,然后对于抗体$\pi \in P'$,设$\text{clone}(\pi)$经过变异后其中亲和度最高的抗体为

π^b,若$F(\pi^b) > F(\pi)$,则用π^b取代原抗体π. 克隆抑制过程实现了对亲和度较低的抗体的抑制,使得优质的变异抗体进入下一代种群.

2) 种群刷新. 用随机生成的新抗体取代种群P'中亲和度较低的$M[\text{pref} \times N_p]$个抗体,其中$\text{pref} = 0.35$称为种群刷新比例. 种群刷新实现了对$P'$中亲和度低的抗体的刷新.

由于该算子的输入包含了原始种群P,克隆选择算法中隐含了最优抗体保留机制,因此种群更新过程中最优解不会变差.

2.4.5　混合克隆选择算法

本文提出的混合克隆选择算法的具体步骤如下.

step 1: 初始化时,随机生成N_p个抗体组成种群P,设置最大迭代次数Φ,迭代次数$\phi = 1$;

step 2: 分别利用式(16)和(18)计算P中每个抗体的激励度和克隆规模,调用克隆算子生成克隆种群P_c;

step 3: 对P_c中所有抗体调用变异算子,生成变异种群P_c';

step 4: 调用克隆抑制和种群刷新算子生成下一代种群P',令$P = P'$;

step 5: 利用算法5对P中抗体执行局部搜索,其中$\Psi = 500$,用生成的更优抗体替代原抗体;

step 6: 令$\phi = \phi + 1$,若$\phi \leqslant \Phi$,则返回step 2;

step 7: 输出找到的最优抗体和对应的天线任务分配结果.

3　数字仿真实验

设计了4组对比实验: 第1组实验对比了HCSA的解与精确算法得到的最优解; 第2组实验验证了HCSA各组成部分的有效性; 第3组实验将HCSA与现有算法进行对比; 最后验证了HCSA解码过程引入阈值参数的有效性. 本文以相对百分比(relative percentage value, RPV)作为算法对比的评价指标,有

$$\text{RPV} = \frac{J^{\text{best}} - J}{J^{\text{best}}}. \tag{20}$$

其中: J是某算法的目标函数值, J^{best}是所有算法在同一算例上的最优目标值. 为减少随机因素,本文考虑了最优RPV(bRPV)和平均RPV(aRPV). 所有算法均基于C++编程,运行在具有Intel Core i9-10900K处理器、128 GB内存、环境为Windows 10的PC机.

3.1　实验场景

本文利用软件STK 11.6搭建Walker Delta低轨星座仿真场景,生成卫星-信关站可见弧段,作为算法输入数据. 仿真开始于[2023-02-01 00:00:00.000 UTCG]. 为生成不同规模的算例,只修改星座卫星数量,轨道高度(2 000 km)和轨道倾角(53.8°)固定不变.

在中国境内设置4个信关站,分别位于北京、楚雄、乌鲁木齐和西安,天线最低仰角设为10°. 利用STK进行卫星-信关站可见性分析,通过改变卫星数量等参数,得到不同规模的实验用例. 表2给出了小、中、大3种规模算例中卫星数、天线数、调度周期的取值. 所有算法在每个算例上独立运行10次.

表 2　算例基本信息

算例规模	卫星数m	天线数a	调度周期T	算例种类数
小	5, 10	1, 2	15	2×2 = 4
中	60, 80	3, 5	3 600	2×2 = 4
大	150, 180	7, 9	3 600	2×2 = 4

3.2　在小规模算例上与精确算法的对比

将HCSA与基于CPLEX实现的精确算法分支定界(B&B)算法进行对比, 规定两种算法的最大运行时间为4 800 s,并设定HCSA运行$\Phi = 200$次迭代后终止. 由于CPLEX难以求解规模较大的算例,本实验仅涉及小规模算例. 实验结果如表3所示,可以看出HCSA多次运行均成功获得最优解,表现出良好的稳定性. 尽管在In(5, 1, 15)上B&B能够快速找到最优解,但在算例规模进一步扩大时其求解时间急剧增加,对于In(10, 2, 15), B&B已无法在规定时间内完成求解,而HCSA总能在5 s之内找到最优解,计算效率更高.

表 3　在小规模算例上HCSA与B&B的对比

算例名称	最优解	B&B			HCSA	
		目标值	Gap%	时间/s	aRPV	时间/s
In(5, 1, 15)	25	**25**	0	**0.22**	**0**	0.69
In(5, 2, 15)	29	**29**	0	145.56	**0**	**1.05**
In(10, 1, 15)	41	**41**	0	4 501.27	**0**	**1.73**
In(10, 2, 15)	78	76	72.06	4 800	**0**	**3.34**

3.3　算法有效性验证

HCSA包含3个重要组成部分: 基于分割任务处理冲突、基于有向图最短路径解码、基于自适应邻域选择的局部搜索. 令HCSA-I表示将HCSA的冲突消解方法修改为单一任务选择的版本,将HCSA中解码方式改为随机解码后的算法为HCSA-II, HCSA删除局部搜索算法后的算法为HCSA-III. 通过对比分析验证上述3部分的有效性,本节设定各算法运行$T_{\max} = 100 \times |G| \times a$ ms后终止.

实验结果如表4所示. 可以看出HCSA-I在中大规模算例上与HCSA存在较大差距,表明在任务冲突严重时HCSA-I的冲突消解方法容易造成天线资源浪费, 导致算法性能下降; HCSA-II采用随机解码, 增

表 4 HCSA与其变体算法的对比

算例名称	$\|G\|$	最优目标值	HCSA-I		HCSA-II		HCSA-III		HCSA	
			*b*RPV	*a*RPV	*b*RPV	*a*RPV	*b*RPV	*a*RPV	*b*RPV	*a*RPV
In(5, 1, 15)	12	25	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
In(5, 2, 15)		29	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
In(10, 1, 15)	46	41	0.024	0.024	**0**	0.022	**0**	**0**	**0**	**0**
In(10, 2, 15)		78	**0**	**0**	0.013	0.037	**0**	**0**	**0**	**0**
In(60, 3, 3 600)	130	40 135	0.061	0.067	0.011	0.015	0.020	0.024	**0**	**0.002**
In(60, 5, 3 600)		51 454	0.026	0.028	0.005	0.009	0.018	0.025	**0**	**0.002**
In(80, 3, 3 600)	210	41 415	0.043	0.045	0.020	0.024	0.002	0.004	**0**	**0.001**
In(80, 5, 3 600)		65 724	0.047	0.050	0.026	0.032	0.016	0.019	**0**	**0.004**
In(150, 7, 3 600)	384	96 876	0.036	0.038	0.040	0.045	0.013	0.013	**0**	**0.002**
In(150, 9, 3 600)		118 762	0.032	0.036	0.035	0.039	0.017	0.019	**0**	**0.002**
In(180, 7, 3 600)	421	92 894	0.040	0.041	0.039	0.042	0.018	0.020	**0**	**0.002**
In(180, 9, 3 600)		108 985	0.021	0.022	0.039	0.041	0.013	0.014	**0**	**0.001**

加了天线执行任务的数量,导致天线方向调整频繁、空闲时间延长,因此效果很差;HCSA-III在多数算例上已超越另外两种变体算法,但相比HCSA仍有差距,HCSA基于自适应邻域选择进行局部搜索,具有更强的求解能力,得到了所有算例的最优结果.

3.4 与其他启发式算法的对比

将HCSA与5种启发式算法进行对比,包括最长服务时间算法(MST)[18]、蚁群算法(ACA)[11]、人工蜂群算法(LB-ABC)[13]、基于知识的遗传算法(KB-GA)[19]和改进遗传算法(IGA)[20]. 本节同样设定算法运行$T_{\max}$ ms后终止.

实验结果如表5所示. 不难看出,HCSA明显优于其他算法,在所有算例上都获得了最小的*b*RPV和*a*RPV,具有更强的求解能力和更稳定的性能. 在卫星数60以内的6个算例上,LB-ABC、IGA和KB-GA性能接近,但在算例规模进一步扩大时差距增大,KB-GA性能更好. ACA相比上述3种算法没有明显优势. MST在小规模算例上表现最差,但在规模扩大时性能改善. 在卫星数超过60后,MST明显优于其他4种对比算法,但与HCSA仍具有一定差距.

为了进一步验证HCSA与其他算法的显著性差异,利用Friedman检验对所有算法进行非参数检验,检验依据为表5中的*b*RPV值. Friedman检验获得的渐近显著性p值为6.7×10^{-5},远小于0.01,因此HCSA

表 5 HCSA与现有启发式算法的对比

算例名称	$\|G\|$	LB-ABC		IGA		ACA		KB-GA		MST		HCSA	
		*b*RPV	*a*RPV	*b*RPV	*a*RPV	*b*RPV	*a*RPV	*b*RPV	*a*RPV	*b*RPV	*a*RPV	*b*RPV	*a*RPV
In(5, 1, 15)	12	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	\	**0**	**0**
In(5, 2, 15)		**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	\	**0**	**0**
In(10, 1, 15)	46	0.024	0.024	0.024	0.024	**0**	0.010	0.024	0.024	0.171	\	**0**	**0**
In(10, 2, 15)		**0**	**0**	**0**	0.010	0.077	0.087	**0**	0.029	0.205	\	**0**	**0**
In(60, 3, 3 600)	130	0.084	0.095	0.084	0.103	0.065	0.075	0.074	0.088	0.037	\	**0**	**0.002**
In(60, 5, 3 600)		0.026	0.027	0.031	0.041	0.080	0.090	0.026	0.035	0.050	\	**0**	**0.002**
In(80, 3, 3 600)	210	0.081	0.091	0.070	0.086	0.039	0.045	0.049	0.060	0.009	\	**0**	**0.001**
In(80, 5, 3 600)		0.088	0.094	0.074	0.091	0.093	0.098	0.054	0.068	0.027	\	**0**	**0.004**
In(150, 7, 3 600)	384	0.104	0.109	0.088	0.096	0.062	0.067	0.057	0.065	0.017	\	**0**	**0.002**
In(150, 9, 3 600)		0.078	0.091	0.077	0.083	0.069	0.072	0.052	0.058	0.016	\	**0**	**0.002**
In(180, 7, 3 600)	421	0.112	0.121	0.095	0.102	0.062	0.064	0.057	0.073	0.021	\	**0**	**0.002**
In(180, 9, 3 600)		0.081	0.087	0.060	0.068	0.074	0.075	0.043	0.049	0.022	\	**0**	**0.001**

与对比算法存在显著性差异. 此外, HCSA的秩平均值(1.58)最小, 随后依次是MST (3.17)、KB-GA (3.25)、ACA (4.04)、IGA (4.25)和LB-ABC (4.71), 表明HCSA的优化性能最好. 最后, 选取上述算法的bRPV和aRPV值, 进行显著性水平$\alpha = 0.05$的Wilcoxon符号秩检验, 结果显示HCSA与其他算法相比的p值均小于0.05, 表明HCSA显著优于对比算法.

3.5 阈值参数的有效性验证

为了验证阈值参数的有效性, 本节基于算例In (80, 5, 3 600), 固定其他参数, 只修改阈值参数WT的取值, 统计HCSA的平均目标函数值、标准差和平均运行时间. 本节设定HCSA运行$\Phi = 200$次迭代后终止. 实验结果如表6所示, 其中“\”一行表示未在算法3中施加阈值限制时的结果. 可以看出, 当WT取值增加时, HCSA的运行时间呈递增趋势, 但WT增大到一定程度后, 目标函数值不会出现明显变化. 其原因在于: 随着WT增大被添加到有向图的弧线权重较大, 不会改变天线的任务序列, 只会增加最短路算法的开销. 由于不施加阈值限制可等效为令WT足够大, 上述结果表明, 引入合适的阈值参数能够在保证求解质量的同时有效降低运行时间. “\”一行的结果也验证了这一点.

表 6　WT变化时HCSA的实验结果

WT取值	平均目标函数值	标准差	平均运行时间/s
1 × HT	42 252	15.06	47.91
5 × HT	45 719	19.85	56.61
10 × HT	58 107	70.24	64.41
20 × HT	65 454	139.40	76.79
30 × HT	65 429	143.89	84.28
40 × HT	65 442	122.87	90.35
50 × HT	65 448	140.27	94.31
60 × HT	65 439	101.40	98.88
\	65 491	105.11	105.38

4 结　论

本文提出一种混合克隆选择算法求解低轨星座馈电链路切换问题. 通过分析和验证, 获得以下结论:

1) 基于任务分割的冲突消解方法能有效解决任务冲突, 特别适用于星座规模不断扩张的实际场景.

2) 基于有向图最短路的解码方法能够显著提高求解质量. 解码过程中阈值参数的引入能够在保证求解质量的前提下有效控制算法运行时间.

3) 自适应邻域选择的局部搜索算法能够有效增强算法局部寻优能力, 使算法收敛到质量更好的解.

4) 实验结果表明, 本文所提出算法能够快速收敛到小规模算例的最优解, 同时在大规模算例上的性能表现显著优于文献中的启发式算法.

参考文献(References)

[1] 陈全, 杨磊, 郭剑鸣, 等. 低轨巨型星座网络: 组网技术与研究现状[J]. 通信学报, 2022, 43(5): 177-189.
(Chen Q, Yang L, Guo J M, et al. LEO mega-constellation network: Networking technologies and state of the art[J]. Journal on Communications, 2022, 43(5): 177-189.)

[2] Levchenko I, Xu S Y, Wu Y L, et al. Hopes and concerns for astronomy of satellite constellations[J]. Nature Astronomy, 2020, 4(11): 1012-1014.

[3] Barbulescu L, Watson J P, Whitley L D, et al. Scheduling space-ground communications for the air force satellite control network[J]. Journal of Scheduling, 2004, 7(1): 7-34.

[4] Song Y J, Ma X, Li X J, et al. Learning-guided nondominated sorting genetic algorithm II for multi-objective satellite range scheduling problem[J]. Swarm and Evolutionary Computation, 2019, 49: 194-205.

[5] Zufferey N, Vasquez M. A generalized consistent neighborhood search for satellite range scheduling problems[J]. RAIRO-Operations Research, 2015, 49(1): 99-121.

[6] Luo K P, Wang H H, Li Y J, et al. High-performance technique for satellite range scheduling[J]. Computers & Operations Research, 2017, 85: 12-21.

[7] Zhang Z J, Hu F N, Zhang N. Ant colony algorithm for satellite control resource scheduling problem[J]. Applied Intelligence, 2018, 48(10): 3295-3305.

[8] Xhafa F, Herrero X, Barolli A, et al. A simulated annealing algorithm for ground station scheduling problem[C]. The 16th International Conference on Network-Based Information Systems. Gwangju, 2013: 24-30.

[9] 杜永浩, 邢立宁, 陈盈果, 等. 卫星任务调度统一化建模与多策略协同求解方法[J]. 控制与决策, 2019, 34(9): 1847-1856.
(Du Y H, Xing L N, Chen Y G, et al. Unified modeling and multi-strategy collaborative optimization for satellite task scheduling[J]. Control and Decision, 2019, 34(9): 1847-1856.)

[10] Xhafa F, Sun J Z, Barolli A, et al. Genetic algorithms for satellite scheduling problems[J]. Mobile Information Systems, 2012, 8(4): 351-377.

[11] 金云忠, 高扬. 基于蚁群算法的卫星地面站任务规划方法[J]. 计算机与现代化, 2012(12): 11-15.
(Jin Y Z, Gao Y. Mission planning method of satellite

ground station based on ant colony algorithm[J]. Computer and Modernization, 2012(12): 11-15.)

[12] Chen Y, Zhang D Y, Zhou M Q, et al. Multi-satellite observation scheduling algorithm based on hybrid genetic particle swarm optimization[C]. Advances in Information Technology and Industry Applications. Berlin, Heidelberg: Springer, 2012: 441-448.

[13] Song Y J, Wei L N, Xing L N, et al. Solving satellite range scheduling problem with learning-based artificial bee colony algorithm[C]. International Conference on Bio-Inspired Computing: Theories and Applications. Singapore: Springer, 2022: 43-57.

[14] Hao H C, Jiang W, Li Y J. Improved algorithms to plan missions for agile earth observation satellites[J]. Journal of Systems Engineering and Electronics, 2014, 25(5): 811-821.

[15] Xu Y, Jiao Y Y, Pan X G, et al. An efficient scheduling method for satellite TT&C resources[C]. The 7th International Conference on Big Data and Information Analytics (BigDIA). Chongqing, 2021: 340-349.

[16] 潘昌忠, 罗晶, 周兰, 等. 基于免疫优化的平面Acrobot线性自抗扰鲁棒镇定[J]. 控制与决策, 2020, 35(12): 3053-3058.
(Pan C Z, Luo J, Zhou L, et al. Robust stabilization of planar Acrobot using linear active disturbance rejection control with immune optimization[J]. Control and Decision, 2020, 35(12): 3053-3058.)

[17] Liu J Q, Zhang Z Q, Chen F, et al. A novel hybrid immune clonal selection algorithm for the constrained corridor allocation problem[J]. Journal of Intelligent Manufacturing, 2022, 33(4): 953-972.

[18] Hou Z W, Yi X Q, Zhang Y H, et al. Satellite-ground link planning for LEO satellite navigation augmentation networks[J]. IEEE Access, 2019, 7: 98715-98724.

[19] Song Y J, Xing L N, Wang M Y, et al. A knowledge-based evolutionary algorithm for relay satellite system mission scheduling problem[J]. Computers & Industrial Engineering, 2020, 150: 106830.

[20] Dai C Q, Li C, Fu S, et al. Dynamic scheduling for emergency tasks in space data relay network[J]. IEEE Transactions on Vehicular Technology, 2021, 70(1): 795-807.

[21] de Castro L N, Von Zuben F J. Learning and optimization using the clonal selection principle[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(3): 239-251.

作者简介

任思达 (1997–), 男, 博士生, 从事卫星调度、智能优化算法等研究, E-mail: rensida@stu.xjtu.edu.cn;

冯彦翔 (1987–), 男, 副教授, 博士, 从事空间信息系统优化设计、航天测控等研究, E-mail: fengyanxiang@xjtu.edu.cn;

陈炜 (1984–), 男, 讲师, 硕士, 从事智能优化算法、软件工程等研究, E-mail: rshchchw@hebau.edu.com;

张广辉 (1981–), 男, 副教授, 博士, 从事柔性制造系统控制与优化调度的研究, E-mail: zhangguanghui@stu.xjtu.edu.cn;

杨宜康 (1974–), 男, 教授, 博士, 从事空间信息系统优化设计、航天测控与星间链路等研究, E-mail: yangyk74@mail.xjtu.edu.cn.