

文章编号:1671-6833(2018)05-0011-11

考虑多滑动窗口的中继卫星调度模型及启发式算法

何敏藩<sup>1</sup>, 朱燕麒<sup>2</sup>, 贾学卿<sup>3</sup>

(1. 佛山科学技术学院 数学与大数据学院, 广东 佛山 528000; 2. 北京市遥感信息研究所, 北京 100085; 3. 国防科技大学 电子科学学院, 湖南 长沙 410073)

**摘 要:** 高效的调度方法是提高中继卫星系统应用效能的关键. 中继卫星调度主要是根据用户的任务申请, 科学合理分配中继卫星系统资源, 最大限度地满足各项任务需求, 为中继卫星系统编制最优工作计划. 考虑中继业务中的多滑动时间窗口需求, 构建了中继卫星调度问题的数学规划模型, 以最大化任务完成率和用户期望满足度为目标, 以任务需求约束、资源使用约束作为约束条件. 设计了基于时间自由度的启发式算法, 该算法包括了任务时间自由度评价、任务资源匹配、任务插空和资源更新 4 个算子. 最后, 通过大规模仿真实验验证了算法的有效性.

**关键词:** 中继卫星; 调度; 启发式算法; 优化

**中图分类号:** V57      **文献标志码:** A      **doi:**10.13705/j.issn.1671-6833.2018.05.020

0 引言

跟踪与数据中继卫星(tracking and data relay satellite, TDRS), 简称为中继卫星<sup>[1-2]</sup>, 主要为中、低轨道的航天器提供数据中继、连续跟踪与轨道测控服务. 中继卫星的出现是 20 世纪航天测控领域的重大突破, 其“天基”设计思想从根本上解决了对中、低轨道航天器测控通信的高覆盖率问题, 减少了地面测控站的数量, 在航天活动中发挥着越来越重要的作用, 也成为未来航天器发展趋势<sup>[3-5]</sup>. 我国的中继卫星系统主要承担载人航天保障、登月计划保障、航天器数据传输以及航天器测控任务, 其中前两项属于频率较低的重大任务, 后两项为常规任务. 对于载人航天保障等重大任务而言, 其执行优先级较高, 中继卫星系统将在任务全过程对其优先提供服务, 因此在中继卫星应用模式及调度方法方面不需要较为复杂的机制. 随着我国国家利益的扩展以及航天器数量的增加, 需要执行大量海外任务. 航天器获取的数据主要通过中继卫星系统实现实时传输和利用. 这使得航天器数据传输与测控的任务需求与日俱增, 用户规模庞大, 调度方案的质量将直接影响任务完成的效果以及中继卫星的使用效能, 这对中继卫星资源的合理调度提出了更

高的要求<sup>[6]</sup>. 需要进一步深入考虑中继卫星应用与用户需求的实际情况, 研究符合实际的中继卫星调度模型以及快速稳定的求解算法.

卫星调度问题是一个具有实践和科学意义的优化问题, 国内外取得了不少的研究成果<sup>[7-9]</sup>. 开彩红等<sup>[10]</sup>针对中继卫星单址天线调度问题, 建立了调度约束规划模型, 提出基于人工蜂群(ABC)算法的中继卫星任务调度算法, 并通过仿真数据分析验证了算法的合理性和有效性. 王志淋等<sup>[11]</sup>借鉴时间窗约束的车辆路径问题理论, 提出天线准备时间变长的规划优化问题(LOPVAPT)模型, 通过优化天线的扫描路径, 实现天线准备时间的动态最优取值, 并基于蚁群算法进行了实验验证. 林鹏等<sup>[12]</sup>面向空间网络中任务发生的时间和空间随机性, 建立中继卫星系统服务模型, 并提出了一种动态的空间任务调度方法. 利用空间网络拓扑动态变化的特点, 结合空间任务的多优先级和容忍延时特性, 提出一种基于种群联合进化的资源分配算法. 顾中舜<sup>[13]</sup>研究了中继卫星动态调度问题, 建立了中继卫星动态调度框架, 分析了中继卫星工作的约束条件, 提出了任务完成评价指标, 创造性地提出完成任务的收益与完成任务时间窗口的选择有关, 考

虑了任务的具体执行时间在窗口中的分布情况.陈理江等<sup>[14]</sup>研究了基于时间灵活度的中继卫星调度算法,在已有的调度方案基础上,通过调度插空和任务交换优化原调度方案,该算法机制比较简单,且可以动态调整.

张彦等<sup>[15]</sup>在 TDRS 动态调度问题求解算法研究中,设计了动态扩展/删除树搜索算法(dynamic extended/delete tree search, DEDTS),该算法在搜索过程中利用基于动态扰动测度的目标值控制删除任务和调整任务的时机,从而更好地体现实际应用的需求.智能优化算法在求解复杂优化问题上表现出优异性能<sup>[16]</sup>,近年来,蚁群算法(ant colony optimization, ACO)、遗传算法(genetic algorithm, GA)、模拟退火(simulated annealing, SA)等智能优化算法在求解组合优化问题方面显示了较强的能力<sup>[17-19]</sup>,在中继卫星调度问题中也得到了一定的应用<sup>[20]</sup>.但是从文献的仿真实验来看,对于中继卫星调度问题,智能优化算法只适用于中、小规模场景的求解,面对多中继卫星多用户航天器大规模任务申请时会出现“维数灾”问题,导致智能优化算法性能迅速下降,求解时间也将大大超出实际调度工作要求.

顾中舜<sup>[13]</sup>利用蚁群算法求解中继卫星初始调度模型,并与基于遗传算法和模拟退火算法的计算结果进行了比较,结果表明,蚁群算法在求解时间和求解精度上都明显优于另外两种算法.方炎申等<sup>[21]</sup>采用基于有效的基因路径的遗传算法来实现中继卫星单址链路任务调度,分析了调度问题中时间窗口的特性,对基本遗传算法进行改进,引入有效基因概念,应用结果表明,采用基于有效基因路径表示的遗传算法求解是合理的.顾中舜<sup>[13]</sup>比较了蚁群算法、遗传算法和模拟退火算法求解中继卫星初始调度模型的计算结果,最后得出由于模拟退火算法有限度地接受劣解,可以跳出局部最优解,但有计算量过大和控制参数过多的缺点.开彩红等<sup>[10]</sup>提出了一种基于人工蜂群(artificial bees colony, ABC)算法的中继卫星任务编排算法.通过在算法的迭代过程中对每个可行调度方案进行评估、邻域操作寻找局部最优调度方案,从而获得全局最优调度方案.邓博于等<sup>[22]</sup>针对遗传算法容易陷入局部最优和蚁群算法初始信息素匮乏的缺点,提出将遗传和蚁群融合算法应用于中继卫星系统的资源调度问题.通过改进蚁群算

法信息素的定义,利用基于时间窗口序号编码思想,给出中继卫星资源调度约束条件与目标函数并建立数学模型. Liu 等<sup>[23]</sup>研究了基于数据中继卫星的空间网络调度问题,在模型中考虑了天线转动时间. Lin 等<sup>[24]</sup>提出了一种非对称路径重连接方法解决中继系统中的工作调度问题. Deng 等<sup>[25]</sup>提出初始调度与动态调度相结合求解中继卫星系统的调度问题.

笔者针对中继卫星单址天线调度问题进行研究,主要贡献包括:着眼中继卫星系统应用实际以及用户需求特点,面向多中继卫星多用户航天器大规模调度问题,扩展用户申请的自主性和灵活性,设计基于多滑动窗口用户申请的中继卫星应用模式,允许用户提交多个备选服务窗口,指定天线偏好,使得模型更加符合实际情况;详细分析该应用模式下中继卫星单址天线调度问题的优化目标和约束条件并建立相应的优化调度模型;设计相应的启发式算法求解调度问题;最后通过仿真实验验证该调度方法的有效性.

## 1 中继卫星调度模型

笔者对于中继卫星单址天线调度问题的研究,面向的是多中继卫星多用户航天器的大规模调度问题,该问题有两个主要的输入:任务申请和中继卫星资源.任务申请即是各用户航天器根据自身需求,向中继卫星相关管理机构提出的对某段时间窗口的占用申请.中继卫星资源主要指的是中继链路,由于笔者研究的是单址天线的调度问题,中继卫星可以搭载多个单址天线,一个单址天线在同一时刻只能提供一条中继链路,即在同一时刻只能执行一项任务.因此可以将中继卫星单址天线与链路等同,作为基本资源单位.本研究中,中继卫星资源输入信息包括多颗中继卫星搭载的多个单址天线的可用时间窗口,以及各中继卫星单址天线对各用户航天器的可见时间窗口.其中,可用时间窗口由中继卫星相关管理机构发布,可见时间窗口由中继卫星和用户航天器轨道参数决定.

中继单址天线调度问题的输出主要是针对每个中继卫星单址天线的调度方案,可以从任务和天线两个角度进行定义.从任务角度分析,中继单址天线调度问题的输出即是确定每项任务是否成功调度,若成功调度,确定该任务由哪条天线对其进行服务,以及服务的开始时刻、结束时刻、持续时间等.从中继卫星天线角度分析,中继单址天线

调度问题的输出即是对每个天线确定周期内的工作计划,具体包括何时开始对哪个用户航天器执行任务,以及任务的结束时刻、持续时间等。

1.1 模型假设与符号说明

首先对数学模型中将用到的符号进行说明,如表 1 所示。

表 1 集合符号定义  
Tab. 1 Definition of used notations

集合符号	含义说明
$R$	中继卫星单址天线集合,中继卫星单址天线 $r \in R$
$T$	已提交任务集合,任务 $t \in T$
$K_t$	任务 $t$ 的备选服务时间窗口集合,任务 $t$ 的第 $k$ 个备选服务时间窗口 $t_k \in K_t$
$L_t$	任务 $t$ 的用户航天器与中继卫星单址天线间的链路集合,任务 $t$ 的用户航天器与中继卫星单址天线链路 $tr \in L_t$
$J_{t,r}$	任务 $t$ 的用户航天器与中继卫星单址天线 $r$ 可见时间窗口集合,可见时间窗口 $tr_j \in J_{t,r}$
$J_r$	中继卫星单址天线 $r$ 可用时间窗口集合,可用时间窗口 $r_l \in J_r$
$adjust$	每项任务开始前中继卫星单址天线对准时间
$rec$	每项任务完成后中继卫星单址天线恢复调整时间
$r_{t,t_k}$	任务 $t$ 的备选服务时间窗口 $t_k$ 指定使用的中继卫星单址天线, $r_{t,t_k} \in R, tr_{t,t_k} \in L_t$
$\bar{r}_{t,t_k}$	任务 $t$ 的备选服务时间窗口 $t_k$ 偏好使用的中继单址卫星天线, $\bar{r}_{t,t_k} \in R, t \bar{r}_{t,t_k} \in L_t$
$start_{t,t_k}$	任务 $t$ 的备选服务时间窗口 $t_k$ 的开始时刻
$forward_{t,t_k}$	任务 $t$ 的备选服务时间窗口 $t_k$ 的开始时刻可前移时段
$backward_{t,t_k}$	任务 $t$ 的备选服务时间窗口 $t_k$ 的开始时刻可后移时段
$d_{t,t_k}$	任务 $t$ 的备选服务时间窗口 $t_k$ 的期望服务时长
$dshort_{t,t_k}$	任务 $t$ 的备选服务时间窗口 $t_k$ 的最短服务时长
$start_{tr_j}$	任务 $t$ 的用户航天器与中继卫星单址天线 $r$ 可见时间窗口 $tr_j$ 开始时刻
$end_{tr_j}$	任务 $t$ 的用户航天器与中继卫星单址天线 $r$ 可见时间窗口 $tr_j$ 结束时刻
$start_{r_l}$	中继卫星单址天线 $r$ 可用时间窗口 $r_l$ 开始时刻
$end_{r_l}$	中继卫星单址天线 $r$ 可用时间窗口 $r_l$ 结束时刻
$x_{t,t_k,r,tr_j,r_l}$	0 ~ 1 变量,表示任务 $t$ 是否选择在备选服务时间窗口 $t_k$ ,中继卫星单址天线 $r$ ,可见时间窗口 $tr_j$ ,可用时间窗口 $r_l$ 内完成
$start_{t,t_k,r,tr_j,r_l}$	任务 $t$ 的备选服务时间窗口 $t_k$ 在中继卫星单址天线 $r$ ,可见时间窗口 $tr_j$ ,可用时间窗口 $r_l$ 下的实际任务开始时刻
$time_{t,t_k,r,tr_j,r_l}$	任务 $t$ 的备选服务时间窗口 $t_k$ 在中继卫星单址天线 $r$ ,可见时间窗口 $tr_j$ ,可用时间窗口 $r_l$ 下的实际任务服务时长
$E_{t,r}$	任务 $t$ 在中继卫星单址天线 $r$ 中实际执行的时间段

1.2 中继卫星单址天线周期调度问题的数学规划模型

中继卫星单址天线调度问题,实质上是要建立中继卫星资源与各任务之间的映射关系.其目的是使中继卫星资源在一个调度周期内完成的任务能够最大程度地满足用户的需求.因此,必须对任务和中继卫星资源进行统一管理,通过调度规划,为各项任务分配相应的资源,使调度方案在满足各项约束的条件下达到预期的效果。

在笔者介绍的中继卫星应用模式下,中继卫星单址天线调度问题就是对每项任务申请确定以下 6 个维度的信息:选择在哪个备选服务时间窗口执行、选择哪个中继卫星单址天线执行、选择该中继卫星单址天线的哪个可用时间窗口执行、选择该中继卫星单址天线与任务所属用户航天器的

哪个可见窗口执行、实际任务开始时刻以及实际任务服务时长,因此,决策变量包括  $x_{t,t_k,r,tr_j,r_l}$ 、 $start_{t,t_k,r,tr_j,r_l}$  和  $time_{t,t_k,r,tr_j,r_l}$ 。

$x_{t,t_k,r,tr_j,r_l}$  为 0 ~ 1 变量,其值确定以下 4 个维度的信息:选择在哪个备选服务时间窗口执行、选择哪个中继卫星单址天线执行、选择该中继卫星单址天线的哪个可用时间窗口执行、选择该中继卫星单址天线与任务所属用户航天器的哪个可见窗口执行。

优化目标反映了优化问题研究的优化对象的本质.针对中继卫星单址天线调度问题,建立两个优化目标,其中任务完成率最大化是首要优化目标,用户期望满足度最大化是次要优化目标。

(1)任务完成率最大化

在大规模任务申请条件下,不可能所有的任

务申请都得到满足,任务完成率就成为评价调度方案的重要指标,是调度模型的主要优化目标.

$$\text{Max } f_1 = \frac{\sum_{t \in T} \sum_{t_k \in K_t} \sum_{r \in R} \sum_{t_{rj} \in J_{t,r}} \sum_{r_{lj} \in J_r} x_{t,t_k,r,t_{rj},r_{lj}}}{[T]}. \quad (1)$$

(2) 用户期望满足度最大化

最大程度满足用户任务申请的偏好或者

$$\text{Max } f_2 = \frac{\sum_{t \in T} \sum_{t_k \in K_t} \sum_{r \in R} \sum_{t_{rj} \in J_{t,r}} \sum_{r_{lj} \in J_r} ((x_{t,t_k,r,t_{rj},r_{lj}} \cdot \bar{r}_{t,t_k} = r) \vee (x_{t,t_k,r,t_{rj},r_{lj}} \cdot r_{t,t_k} = r)) \wedge (time_{t,t_k,r,t_{rj}} = d_{t,t_k})}{[T]}. \quad (2)$$

中继卫星调度问题主要包含两类约束:任务需求约束和资源使用约束. 任务需求约束主要是用户根据其需求,提出的任务申请中包含的约束.

(1) 任务的执行约束. 在笔者提出的中继卫星应用模式中,允许用户根据实际需求提交多个备选服务时间窗口,在生成调度方案时,仅选择其中的一个备选服务时间窗口进行调度,不可对任务进行重复安排,并且每项任务仅选择一个中继卫星单址天线执行. 根据该约束得到 Constraint 1~2.

Constraint 1: 每个任务仅选择一个备选服务时间窗口,

$$\sum_{t_k \in K_t} x_{t,t_k,r,t_{rj},r_{lj}} \leq 1. \quad (3)$$

Constraint 2: 每个任务仅选择一个中继卫星单址天线执行,

$$\sum_{r \in R} x_{t,t_k,r,t_{rj},r_{lj}} \leq 1. \quad (4)$$

(2) 任务的服务时间窗口约束. 由于任务需求具有很强的时效性,例如遥感卫星获取的某些敏感数据必须通过中继卫星在用户申请的时间段内回传到地面站,因此,中继卫星对任务的服务时间窗口必须在相应的备选服务时间窗口范围内,过早或过晚都会影响任务执行效果,导致任务失败,根据该约束得到 Constraint 3~4.

Constraint 3: 每个任务的实际开始时刻不早于所在备选服务时间窗口的最早开始时刻,

$$start_{t,t_k,r,t_{rj},r_{lj}} \geq (start_{t,t_k} - forward_{t,t_k}) \cdot x_{t,t_k,r,t_{rj},r_{lj}}. \quad (5)$$

Constraint 4: 每个任务的实际开始时刻不晚于所在备选服务时间窗口的最晚开始时刻,

$$start_{t,t_k,r,t_{rj},r_{lj}} \cdot x_{t,t_k,r,t_{rj},r_{lj}} \leq start_{t,t_k} + backward_{t,t_k}. \quad (6)$$

期望,可以提高调度方案的服务水平,有利于提升任务执行的质量和效果. 因此用户期望满足度是调度模型的另一个优化目标. 在笔者的中继卫星应用模式下,用户期望得到满足需要以下两个条件:采用任务备选服务时间窗口指定或偏好的中继卫星天线;采用任务备选服务时间窗口的期望服务时长.

(3) 任务的服务天线约束. 如果选择的任务备选服务时间窗口指定了服务的中继卫星天线,则必须使用指定的中继卫星天线为该项目进行服务,使用其他天线则无法满足用户需求,视为任务失败,根据该约束得到 Constraint 5.

Constraint 5: 对于选择指定天线的备选服务时间窗口的任务,执行天线必须满足指定要求,

$$\forall ((r_{t,t_k} \neq \emptyset) \wedge (x_{t,t_k,r,t_{rj},r_{lj}} \neq 0)), r = r_{t,t_k}. \quad (7)$$

(4) 任务的服务时长约束. 选择的任务备选服务时间窗口包含期望服务时长和最短服务时长,实际调度方案中的任务服务时长应不大于期望服务时长,不小于最短服务时长,根据该约束得到 Constraint 6~7.

Constraint 6: 每个任务的实际服务时长不大于备选服务时间窗口的期望服务时长,

$$time_{t,t_k,r,t_{rj},r_{lj}} \leq d_{t,t_k}. \quad (8)$$

Constraint 7: 每个任务的实际服务时长不小于备选服务时间窗口的最短服务时长,

$$time_{t,t_k,r,t_{rj},r_{lj}} \geq dshort_{t,t_k}. \quad (9)$$

(5) 中继卫星单址天线能力约束. 一个单址天线在同一时刻只能提供一条中继链路,即在同一时刻只能执行一项任务,并且在任务开始之前,中继卫星天线需要根据预先计算结果,提前将天线转动到合适的角度与用户航天器进行对准操作,天线对准需要占用一定时间. 在任务执行结束之后,中继卫星天线需要一段时间进行恢复调整,为执行下一项任务做好准备. 在进行天线对准操作和天线恢复调整过程中,无法执行任务或进行其他工作,根据该约束得到 Constraint 8.

Constraint 8: 每个中继卫星单址天线任意时刻仅能执行一项工作,

$$\forall r \in R, t1 \in T, t2 \in T, (adjust + E_{t1,r} + rec) \cap$$

$$(adjust + E_{i2,r} + rec) = \emptyset. \quad (10)$$

(6) 可用时间窗口约束. 可用时间窗口由中继卫星相关管理机构发布, 由于任务执行过程不允许中断, 因此中继卫星执行任务相关的所有操作, 包括天线对准、任务执行与天线调整恢复, 必须在中继卫星天线的某个可用时间窗口内完成, 根据该约束得到 Constraint 9 ~ 11.

Constraint 9: 每个任务在中继卫星单址天线的可用时间窗口内执行,

$$\sum_{r_l \in J_r} x_{t,t_k,r,tr_j,r_l} \leq 1. \quad (11)$$

Constraint 10: 每个任务开始前的天线对准开始时刻不早于所在单址天线的可用时间窗口的开始时刻,

$$start_{t,t_k,r,tr_j,r_l} - adjust \geq start_{r_l} \cdot x_{t,t_k,r,tr_j,r_l}. \quad (12)$$

Constraint 11: 每个任务结束后的天线调整恢复结束时刻不晚于所在单址天线的可用时间窗口的结束时刻,

$$(start_{t,t_k,r,tr_j,r_l} + time_{t,t_k,r,tr_j,r_l} + rec) \cdot x_{t,t_k,r,tr_j,r_l} \leq end_{r_l}. \quad (13)$$

(7) 可见时间窗口约束. 可见时间窗口由中继卫星和用户航天器轨道参数决定, 在任务执行过程中必须时刻保持中继卫星天线与用户航天器的直视. 由于任务执行过程不允许中断, 因此中继卫星的任务执行过程必须在中继卫星天线与用户航天器的某个可见时间窗口内完成. 任务执行前的天线对准和任务完成后的天线调整恢复操作, 不要求在可见窗口内完成, 根据该约束得到 Constraint 12 ~ 14.

Constraint 12: 每个任务在中继卫星单址天线与用户航天器的一个可见时间窗口内执行,

$$\sum_{tr_j \in J_{t,r}} x_{t,t_k,r,tr_j,r_l} \leq 1. \quad (14)$$

Constraint 13: 每个任务的实际开始时刻不早于所在单址天线与用户航天器的可见时间窗口的开始时刻,

$$start_{t,t_k,r,tr_j,r_l} \geq start_{tr_j} \cdot x_{t,t_k,r,tr_j,r_l}. \quad (15)$$

Constraint 14: 每个任务的实际结束时刻不晚于所在单址天线与用户航天器的可见时间窗口的结束时刻,

$$(start_{t,t_k,r,tr_j,r_l} + time_{t,t_k,r,tr_j,r_l}) \cdot x_{t,t_k,r,tr_j,r_l} \leq end_{tr_j}. \quad (16)$$

笔者建立的中继卫星单址天线周期调度模型为 NP-hard 问题, 考虑多个滑动的备选服务时间窗口、多个中继卫星单址天线、多个可见时间

窗口和多个可用时间窗口, 问题的解空间随着资源与任务数量的增长呈指数激增且面临着组合爆炸的挑战. 模型具有超高的决策变量维度以及复杂的约束条件, 尚不存在按多项式时间复杂度找到全局最优解的算法, 现有通用算法难以直接求解本模型, 对模型求解算法的设计提出了更高的要求.

## 2 基于时间自由度的启发式算法

基于规则的启发式算法, 具有求解速度快、稳定性高的优点, 能够在较短的时间内给出较优的可行解, 求解中继卫星调度问题的有效算法. 笔者设计的基于时间自由度的启发式算法, 考虑到用户的任务申请可以包含多个备选服务时间窗口, 且备选服务时间窗口有可能指定或不指定中继卫星天线. 依据任务申请包含的备选服务时间窗口数量以及指定天线情况, 对任务的时间自由度进行评价, 根据评价结果决定任务调度的优先次序. 在调度理论中, 该算法属于构造性算法, 即依次对任务进行调度, 最终生成较优的可行调度方案.

基于时间自由度的启发式算法主要包括以下 4 个算法模块: ①任务时间自由度评价; ②任务资源匹配; ③任务插空; ④资源刷新. ①是根据任务的备选服务时间窗口数量及指定天线情况, 评价任务的时间自由度并决定优先次序. ②是根据任务提交的备选服务时间窗口信息, 为任务匹配当前可用时段资源. ③是基于启发式规则, 根据②中任务各备选服务时间窗口与当前可见时间窗口以及当前天线可用时间窗口的匹配结果, 为当前任务安排合适的调度方案. ④是在安排新的任务之后, 对现有可见时间窗口以及天线可用时间窗口进行刷新, 删除已被占用的资源.

### 2.1 任务时间自由度评价

根据任务的备选服务时间窗口数量及指定天线情况, 评价该任务的时间自由度. 一般来说, 提交的备选服务时间窗口数量越多, 不指定天线的备选服务时间窗口数量越多, 任务的时间自由度越高, 任务调度的难度越小, 可以使其获得较低的优先次序, 在算法运行的靠后阶段进行调度.

具体的评价方式为: 首先遍历任务集合  $T$ , 获取最大备选服务时间窗口数量  $win_{\max} = \max\{[K_t], t \in T\}$ ; 然后依次分析任务集合  $T$  中的任务, 获取当前任务  $t$  的备选服务时间窗口数量

$win_t = \lfloor K_t \rfloor$ , 当前任务  $t$  指定天线的备选服务窗口数量  $win_{assign}$ ; 最后对当前任务  $t$  进行时间自由度评价, 评价公式为:

$$score_t = (win_{max} + 1 - win_t) + (win_{max} - win_t + win_{assign}). \quad (17)$$

对其进行简化, 即

$$score_t = 2 \times (win_{max} - win_t) + win_{assign} + 1. \quad (18)$$

式中:  $score_t$  的取值范围为  $[1, 2 \times win_{max}]$ , 任务  $t$  的时间自由度评价  $score_t$  取值越大, 表明任务  $t$  的时间自由度越低, 将取得较为靠前的调度次序.

## 2.2 任务资源匹配

对于当前处理的任务  $t$ , 遍历其提交的备选服务时间窗口  $K_t$ , 与现有资源进行比对, 得到可以安排的可用时段资源. 任务资源匹配需要与当前可见时间窗口和当前天线可用时间窗口分别进行比对, 将匹配得到的可用时段资源分成两类存放: 一类是指定天线的备选服务时间窗口匹配得到的可用时段资源集合, 记为  $taskresource1$ ; 另一类是不指定天线的备选服务时间窗口匹配得到的可用时段资源集合, 并记录其偏好信息, 记为  $taskresource2$ .

任务资源匹配方法如图 1 所示, 其中  $dtime_{t,t_k}$  是任务  $t$  的备选服务时间窗口  $t_k$  在任务资源匹配时采用的服务时长.

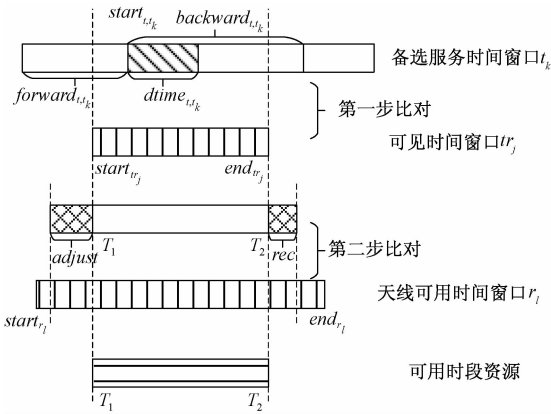


图 1 任务资源匹配方法

Fig. 1 The task-resource matching method

对于当前备选服务时间窗口  $t_k$ , 若  $t_k$  指定中继卫星单址天线, 则首先遍历其指定天线  $r_{t,t_k}$  与该任务用户航天器当前可见时间窗口集合  $J_{t,r}$ . 对于  $J_{t,r}$  中的可见时间窗口  $tr_j$ , 若满足条件

$$\begin{cases} start_{tr_j} \geq start_{t,t_k} + backward_{t,t_k} + dtime_{t,t_k} \\ end_{tr_j} \leq start_{t,t_k} - forward_{t,t_k} \end{cases}, \quad (19)$$

则可见时间窗口  $tr_j$  与当前备选服务时间窗口  $t_k$  无交集, 可直接排除; 对于与  $t_k$  有交集的可见时间窗口  $tr_j$ , 判断其是否满足以下条件:

$$\begin{cases} T_1 = \max \{ start_{tr_j}, start_{t,t_k} - forward_{t,t_k} \} \\ T_2 = \min \{ end_{tr_j}, start_{t,t_k} + backward_{t,t_k} + dtime_{t,t_k} \} \\ T_2 - T_1 \geq dtime_{t,t_k} \end{cases}, \quad (20)$$

若满足该条件, 则可见时间窗口  $tr_j$  可用, 继续进行下一步比对. 遍历指定天线  $r_{t,t_k}$  的当前可用时间窗口集合  $J_r$ , 对于  $J_r$  中的可用时间窗口  $r_l$ , 若满足条件:

$$\begin{cases} start_{r_l} \leq T_1 - adjust \\ end_{r_l} \geq T_2 + rec \end{cases}, \quad (21)$$

则可用时间窗口  $r_l$  可用. 该步比对加入了任务执行前的天线对准和任务完成后的天线恢复调整时间. 记可用时段资源开始时刻为  $T_1$ , 可用时段资源结束时刻为  $T_2$ , 并同时记录可用时段资源所在中继卫星单址天线, 记入  $taskresource1$ .

对于未指定天线的备选服务时间窗口  $t_k$ , 匹配时要遍历中继卫星单址天线集合  $R$ , 对于某个中继卫星单址天线  $r$ , 其具体匹配方法与上述方法相同, 将备选服务时间窗口  $t_k$  偏好天线信息  $\bar{r}_{t,t_k}$  一并记入  $taskresource2$ .

## 2.3 任务插空

对于当前处理的任务  $t$ , 根据任务资源匹配算法得到的可用时段资源集合  $taskresource1$  和  $taskresource2$ , 进行任务的安排.

首先判断  $taskresource1$  是否非空, 若其非空, 在其中随机选择一个可用时段资源进行调度, 采用紧前、随机或紧后策略, 转入算法 (2.4) 刷新资源. 任务插空方法如图 2 所示.

其中, 紧前策略

$$\begin{cases} start_{t,t_k,r,tr_j,r_l} = T_1 \\ time_{t,t_k,r,tr_j,r_l} = dtime_{t,t_k} \\ E_{t,r} = [start_{t,t_k,r,tr_j,r_l}, start_{t,t_k,r,tr_j,r_l} + time_{t,t_k,r,tr_j,r_l}] \end{cases}. \quad (22)$$

随机策略

$$\begin{cases} start_{t,t_k,r,tr_j,r_l} = T_1 + random() \cdot (T_2 - T_1 - dtime_{t,t_k}) \\ time_{t,t_k,r,tr_j,r_l} = dtime_{t,t_k} \\ E_{t,r} = [start_{t,t_k,r,tr_j,r_l}, start_{t,t_k,r,tr_j,r_l} + time_{t,t_k,r,tr_j,r_l}] \end{cases}. \quad (23)$$

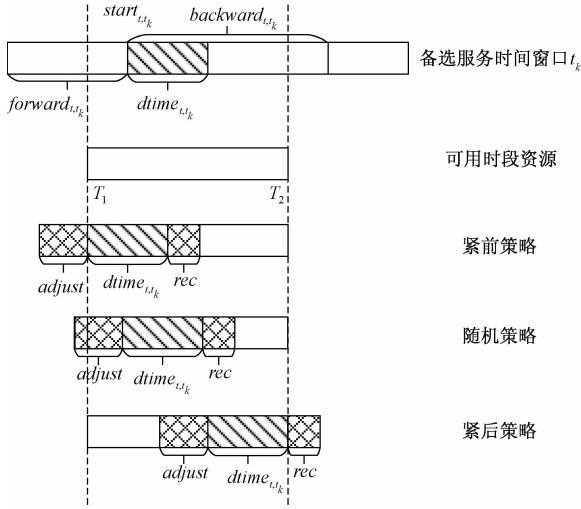


图2 任务插空方法

Fig.2 Task insertion method

$random()$  为随机生成的  $[0,1]$  的随机数.

紧后策略

$$\begin{cases} start_{t,t_k,r,tr_j,r_l} = T_2 - dtime_{t,t_k} \\ time_{t,t_k,r,tr_j,r_l} = dtime_{t,t_k} \\ E_{t,r} = [start_{t,t_k,r,tr_j,r_l}, start_{t,t_k,r,tr_j,r_l} + time_{t,t_k,r,tr_j,r_l}] \end{cases} \quad (24)$$

若  $taskresource1$  为空集,判断  $taskresource2$  的情况,若其非空,遍历其中的可用时段资源,找出其中天线偏好非空且与可用时段资源所在天线相同的时段.如果存在天线偏好非空且与可用时段资源所在天线相同的时段,在其中随机选择一个进行调度,采用紧前、随机或紧后策略,转入资源刷新操作.若不存在天线偏好非空且与可用时段资源所在天线相同的时段,则在  $taskresource2$  中随机选择一个可用时段资源进行调度,采用紧前、随机或紧后策略,转入资源刷新操作.

若  $taskresource2$  也为空集,则将该任务所有备选服务时间窗口采用的服务时长由期望服务时长改为最短服务时长,转入任务资源匹配操作,重新进行任务资源匹配,得到该任务的  $taskresource1$  和  $taskresource2$ ,并重复上述过程.

若  $taskresource1$  和  $taskresource2$  仍均为空集,则认为当前资源无法完成该任务的调度,当前任务调度失败.

## 2.4 资源刷新

每当有任务被成功调度,会占用该中继卫星单址天线与本用户航天器以及其他用户航天器的可见时间窗口以及天线可用时间窗口,由于资源的互斥性,需要对可见时间窗口以及

天线可用时间窗口均进行刷新,删除本次调度占用的时段.被占用的时段不仅包括任务的服务时长  $time_{t,t_k,r,tr_j,r_l}$ ,还包括之前的中继卫星天线对准时间  $adjust$  以及随后的恢复调整时间  $rec$ .

对可见时间窗口资源进行刷新时,要遍历该中继卫星单址天线  $r$  对所用户航天器的所有可见时间窗口,对当前处理的可见时间窗口  $tr_j$ ,令

$$\begin{cases} T_3 = start_{t,t_k,r,tr_j,r_l} - adjust \\ T_4 = start_{t,t_k,r,tr_j,r_l} + time_{t,t_k,r,tr_j,r_l} + rec \end{cases} \quad (25)$$

则本次调度所占用的时段与可见时间窗口  $tr_j$  可能存在的情形如图3所示.

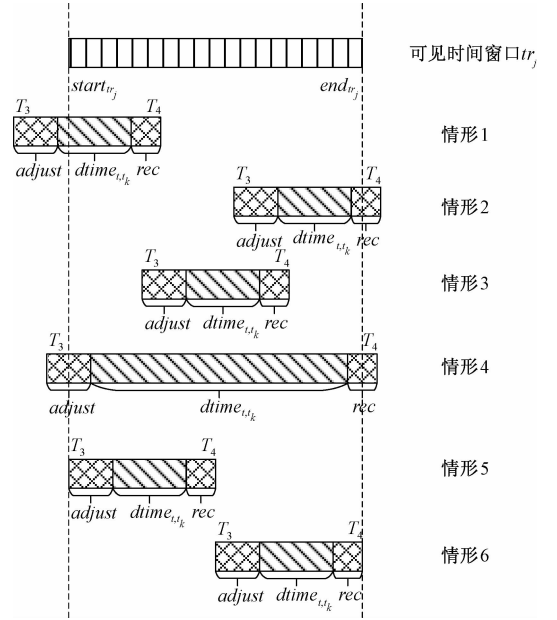


图3 可见时间窗口资源删除及刷新方法

Fig.3 Method for deleting and updating visible time-windows

针对不同情况的刷新方法如表2所示.

表2 针对不同情况的资源刷新规则

Tab.2 Resource update rules under different conditions

情形分类	原可见时间窗口刷新结果
$T_3 < start_{tr_j} < T_4 < end_{tr_j}$	$[T_4, end_{tr_j}]$
$start_{tr_j} < T_3 < end_{tr_j} < T_4$	$[start_{tr_j}, T_3]$
$start_{tr_j} < T_3 < T_4 < end_{tr_j}$	$[start_{tr_j}, T_3], [T_4, end_{tr_j}]$
$T_3 \leq start_{tr_j} < end_{tr_j} \leq T_4$	$\emptyset$
$T_3 = start_{tr_j}, T_4 < end_{tr_j}$	$[T_4, end_{tr_j}]$
$start_{tr_j} < T_3, end_{tr_j} = T_4$	$[start_{tr_j}, T_3]$

天线可用时间窗口资源删除及刷新方法与上述方法相同,在此不再赘述.

### 3 仿真实验

#### 3.1 实验设置

实验平台为配置 3.29 GHz Intel Core CPU、8 GB 内存、Windows 7 操作系统的 PC 机,算法在 Matlab 2014 中编程实现.由于采用基于多滑动窗口用户申请的中继卫星应用模式,没有标准测试集可供调用,为完成仿真实验,首先进行中继卫星资源与任务需求场景仿真,生成所需应用场景.在此基础上,对笔者所提算法进行测试.

中继卫星相关管理机构下属 4 颗中继卫星,每颗中继卫星载有 1 副中继卫星单址天线用于执行常规任务,面向的用户部门共下属 20 颗用户航天器.

根据日常工作安排,中继卫星相关管理机构于 2017 年 11 月 25 日发布 2017 年 12 月 1 日 0 时~2017 年12 月 7 日 0 时的可用中继卫星资源,各用户部门可根据自身工作需要与所属航天器工作情况于 2017 年 11 月 28 日前进行任务申请,允许每项任务申请提交 1~3 个备选服务时间窗口.最终收到 500 个任务申请,中继卫星相关管理机构于 2017 年 11 月 29 日完成周期调度工作,在周期调度方案生成后,2017 年 11 月 30 日有 3 个临时任务申请到达.任务开始前中继卫星单址天线对准时间 10 min,任务结束后中继卫星单址天线恢复调整时间 4 min.

针对以上假定的应用场景,将相关参数整理如表 3 所示.

表 3 应用场景参数设置

Tab.3 The parameter configuration of the application scenario

场景参数	取值
调度周期开始时刻	2017 年 12 月 1 日 0 时
调度周期结束时刻	2017 年 12 月 7 日 0 时
中继卫星单址天线数量	4
用户航天器数量	20
任务申请数量	500
最大备选服务时间窗口数量	3
临时任务数量	3
中继卫星单址天线对准时间	10 min
中继卫星单址天线恢复调整时间	4 min

中继卫星资源主要包括中继卫星单址天线的可用时间窗口,以及各中继卫星单址天线对各用户航天器的可见时间窗口,需仿真相关数据用于后续实验.本案例将 4 个中继卫星单址天线的可用时间窗口设置为 2017 年 12 月 1 日 0 时至 2017 年 12 月 7 日 0 时均可用.对于各中继卫星单址天线对各用户航天器的可见时间窗口,笔者利用 STK 软件设置了相关场景进行仿真,获取相关数据.

#### 3.2 计算结果

利用基于时间自由度的启发式算法对上述案例场景进行测试,其中 1~10 项任务的调度方案如表 4 所示.

表 4 基于时间自由度的启发式算法部分调度结果

Tab.4 Part of the scheduling results of the heuristic algorithm based on time freedom degree

任务编号	所属航天器	是否安排	天线编号	开始时刻	结束时刻	持续时间/s
1	4	1	2	[2017,12,5,15,46,46]	[2017,12,5,15,58,31]	0.008 160 526
2	19	1	4	[2017,12,2,17,34,57]	[2017,12,2,17,52,51]	0.012 437 202
3	20	1	2	[2017,12,2,23,15,28]	[2017,12,2,23,24,53]	0.006 535 57
4	1	1	4	[2017,12,6,10,47,18]	[2017,12,6,11,0,51]	0.009 405 561
5	16	1	2	[2017,12,1,9,10,44]	[2017,12,1,9,21,45]	0.007 642 144
6	17	-1	-1	-1	-1	-1
7	16	-1	-1	-1	-1	-1
8	18	1	3	[2017,12,1,17,21,48]	[2017,12,1,17,40,29]	0.012 972 767
9	5	1	3	[2017,12,2,14,13,50]	[2017,12,2,14,25,25]	0.008 045 782
10	10	1	2	[2017,12,2,22,48,18]	[2017,12,2,22,59,18]	0.007 637 625

表中信息值 -1 表示该项任务由于资源不足未成功调度,时间段数值的单位为天.

算法的主要评价指标数据如表 5 所示.实验结果表明,算法具有很快的运行速度,能够在 2 s 内完成 500 项任务规模场景的周期调度方案,任务完成率可以达到 76.4%,用户期望满足度达到

50.2%,具备较强的可用性,能够满足调度方案快速生成的要求.

#### 3.3 与最大权重任务最先服务算法对比

可以发现基于时间自由度的启发式算法里使用了 4 个灵活的算子,包括任务时间自由度评价、任务资源匹配、任务插空和资源更新,保障算法



表 5 基于时间自由度的启发式算法性能指标

Tab.5 The performance criterion of the heuristic algorithm based on the time freedom degree

评价指标	实验结果
任务申请数量	500
任务完成数量	382
任务完成率/%	76.4
满足用户期望任务数量	251
用户期望满足度/%	50.2
算法运行时间/s	1.797 987

能够快速高效得到中继卫星系统的调度方案. 为了验证算法的有效性, 将其与最大权重任务最先服务算法(highest weight task first service, HWFS)进行比较. 最大权重任务最先服务算法的步骤如下:

Step 1:对待完成的中继任务根据其权重从大到小排序,形成任务队列  $Q$ .

Step 2:然后每次从任务队列  $Q$  中选出权重最大的任务  $t$ .

Step 3:为任务  $t$  安排最佳服务资源,从  $Q$  中删除  $t$ . 如果  $Q = \emptyset$ , 算法结束;否则转 Step 2.

两个算法的计算结果如表 6 所示. 由表 6 中的数据可以看出,在该场景下,HA-TFD 把 HWFS 的任务完成率从 70% 提升到 76.4%,把 HWFS 的用户期望满足度从 43% 提升到 50.2%,该对比实验说明,HA-TFD 是一种更加灵活高效的启发式优化算法.

### 3.4 任务规模对算法的影响

为了分析不同任务规模对算法的影响,本节在任务需求仿真相关参数取值不变的条件下,设置 100、300、500、800、1 200、1 600 共 6 个不同任务申请数量的任务场景,分别对基于时间自由

度的启发式算法和基于冲突消解的随机搜索算法进行测试. 每个任务场景进行 20 次重复实验,对算法性能进行分析,相关数据如表 7 所示.

表 6 算法比较结果

Tab.6 Comparison results

评价指标	HA-TFD	HWFS
任务申请数量	500	500
任务完成数量	382	350
任务完成率/%	76.4	70
满足用户期望任务数量	251	215
用户期望满足度/%	50.2	43
算法运行时间/s	1.797 987	1.58 746

表 7 不同任务规模下的实验结果

Tab.7 The experimental results of different scales of tasks

任务数量	完成任务数	完成率/%	满足期望任务数	平均期望满足度/%	平均运行时间/s
100	96	96.00	75	75.00	0.400 516
300	269	89.67	187	62.33	1.085 394
500	383	76.60	243	48.60	1.613 374
800	503	62.88	298	37.25	2.369 378
1 200	578	48.17	329	27.42	3.115 182
1 600	616	38.50	374	23.38	3.225 724

### 3.5 用户航天器数量对算法的影响

为了分析不同用户航天器数量对算法的影响,在任务需求仿真相关参数取值不变的条件下,仅改变应用场景参数中的用户航天器数量. 设置 4、8、16、20、28、36 共 6 个不同用户航天器数量的任务场景,分别对基于时间自由度的启发式算法和基于冲突消解的随机搜索算法进行测试. 每个任务场景进行 20 次重复实验,实验结果如表 8 所示.

表 8 不同用户航天器数量下的实验结果

Tab.8 Experimental results of different numbers of user Spacecraft

用户航天器数量	任务申请数量	平均任务完成数量	平均任务完成率/%	平均满足用户期望任务数量	平均用户期望满足度/%	平均算法运行时间/s
4	500	360	72.0	225	45.0	0.492 058
8	500	385	77.0	235	47.0	0.794 376
16	500	378	75.6	249	49.8	1.383 058
20	500	387	77.4	242	48.4	1.924 824
28	500	394	78.8	250	50.0	2.284 258
36	500	395	79.0	262	52.4	3.018 492

## 4 结论

分析中继卫星单址天线周期调度问题,考虑多滑动窗口用户申请的中继卫星应用模式,针对中继卫星单址天线周期调度多目标、多约束的特点,建立了数学规划模型. 模型最大化任务完成率

和用户期望满足度,其中任务完成率是最重要的优化目标,用户期望满足度是次要优化目标. 将周期调度中的任务需求约束、资源使用约束作为模型的约束条件. 提出基于时间自由度的启发式算法,算法利用任务申请信息在任务调度之前对任务完成难度进行评估,并以此为依据进行任务调

度优先顺序的安排,提高了启发式算法任务调度的成功率,从而提高了调度方案的质量.笔者创造性地提出基于多滑动窗口用户申请的中继卫星应用模式,大大扩展了用户申请的自主性和灵活性,有利于提高调度方案的质量和中继卫星系统资源的使用效能,对于中继卫星系统实际应用模式的创新具有一定的借鉴意义.基于时间自由度的启发式算法具有求解速度快,算法稳定的优点,适用于应急调度情况和大规模调度场景,其不足之处在于,由于没有引入随机迭代求解策略,求解的质量还有待进一步提高.下一步将继续研究高效智能优化算法求解中继卫星调度问题,提高调度效率和调度方案质量.

参考文献:

- [1] 王永刚. 军事卫星及应用概论[M]. 北京: 国防工业出版社, 2003.
- [2] ZILLIG D, MCOMBER R, HORNE W. Demand access service for TDRSS users [C]//International Communications Satellite Systems Conference, 2006.
- [3] BRANDEL D L, WATSON W A, WEINBERG A. NASA's advanced tracking and data relay satellite system for the years 2000 and beyond[J]. Proceedings of the IEEE. 1990, 78(7): 1141 - 1151.
- [4] TELES J, SAMII M V, DOLL C E. Overview of TDRSS [J]. Advances in space research, 1995, 16(12): 67 - 76.
- [5] 杨红俊. 国外数据中继卫星系统最新发展及未来趋势[J]. 电讯技术, 2016, 56(1): 109 - 116.
- [6] 周志鑫, 吴志刚, 季艳. 空间对地观测技术发展及应用[J]. 中国工程科学, 2008, 10(6): 28 - 32.
- [7] WU G, WANG H, PEDRYCZ W, et al. Satellite observation scheduling with a novel adaptive simulated annealing algorithm and a dynamic task clustering strategy [J]. Computers and industrial engineering, 2017(113): 576 - 588.
- [8] WU G, PEDRYCZ W, LI H, et al. Coordinated planning of heterogeneous earth observation resources[J]. IEEE transactions on systems, man, and cybernetics; systems, 2016, 46(1): 109 - 125.
- [9] WU G, LIU J, MA M, et al. A two-phase scheduling method with the consideration of task clustering for earth observing satellites [J]. Computers and operations research, 2013, 40(7): 1884 - 1894.
- [10] 开彩红, 肖瑶, 方青. 基于人工蜂群算法的中继卫星任务调度研究[J]. 电子与信息学报, 2015, 37(10): 2466 - 2474.
- [11] 王志淋, 李新明. 跟踪与数据中继卫星系统资源调度优化问题[J]. 中国空间科学技术, 2015, 35(1): 36 - 42.
- [12] 林鹏, 晏坚, 费立刚, 等. 中继卫星系统的多星多天线动态调度方法[J]. 清华大学学报(自然科学版), 2015, 55(5): 491 - 496.
- [13] 顾中舜. 中继卫星动态调度问题建模及优化技术研究[D]. 长沙: 国防科学技术大学信息系统与管理学院, 2008.
- [14] 陈理江, 武小悦, 李云峰. 基于时间灵活度的中继卫星调度算法[J]. 航空计算技术, 2006(4): 48 - 51.
- [15] 张彦, 孙占军, 李剑. 中继卫星动态调度问题研究[J]. 系统仿真学报, 2011, 23(7): 1464 - 1468.
- [16] WU G, SHEN X, LI H, et al. Ensemble of differential evolution variants [J]. Information sciences, 2018(423): 172 - 186.
- [17] 常玉林, 汪小淳, 张鹏. 改进蚁群算法在交通分配模型中的应用[J]. 郑州大学学报(工学版), 2017, 38(2): 41 - 44.
- [18] 王俊英, 颜芬芬, 陈鹏, 等. 基于概率自适应蚁群算法的云任务调度方法[J]. 郑州大学学报(工学版), 2017, 38(4): 51 - 56.
- [19] 吴秀丽, 张志强. 求解柔性作业车间调度问题的细菌算法对比及改进[J]. 郑州大学学报(工学版), 2018, 39(3): 34 - 39.
- [20] 赵静, 赵尚弘, 李勇军, 等. 中继卫星资源调度问题研究现状与展望[J]. 电讯技术, 2012, 52(11): 1837 - 1843.
- [21] 方炎申, 陈英武, 顾中舜. 中继卫星调度问题的 CSP 模型[J]. 国防科技大学学报, 2005, 27(2): 6 - 10.
- [22] 邓博于, 赵尚弘, 侯睿, 等. 基于遗传蚁群融合算法的混合链路中继卫星资源调度研究[J]. 红外与激光工程, 2015, 44(7): 2211 - 2217.
- [23] LIU R, SHENG M, XU C, et al. Antenna slewing time aware mission scheduling in space networks [J]. IEEE communications letters, 2017, 21(3): 516 - 519.
- [24] LIN P, KUANG L, CHEN X, et al. Asymmetric path-relinking based heuristics for large-scale job scheduling problem in TDRSS [C]//International Conference on Communications & Networking in China, 2014, 115 - 121.
- [25] DENG B, JIANG C, KUANG L, et al. Two-phase task scheduling in data relay satellite systems [J]. IEEE transactions on vehicular technology, 2018, 67(2): 1782 - 1793.

Scheduling Model and Heuristic Algorithm for Tracking and Data Relay Satellite  
Considering Multiple Slide Windows

HE Minfan<sup>1</sup>, ZHU Yanqi<sup>2</sup>, JIA Xueqing<sup>3\*</sup>

(1. School of Mathematics and Big Data, Foshan University, Foshan 528000, China; 2. Beijing Institute of Remote Sensing Information, Beijing 100085, China; 3. College of Electrical Science and Engineering, National University of Defense Technology, Changsha 410073, China)

**Abstract:** Efficient scheduling algorithm plays a key role in improving the efficacy of tracking and data relay satellite system (TDRS). Scheduling of TDRS aims to scientifically allocate TDRS resources according to the task application information from the users, such that maximal task requirements are met and the optimal activity schedule is made for the TDRS system. The mathematical model is constructed for the TDRS scheduling problem with the consideration of multiple slide windows in real-world requirements. The objective of the model is to maximize the task completion rate and the expectation satisfaction degree of users. The involved constraints include task requirement constraints and resource using constraints. A heuristic algorithm based on time freedom degree is proposed to solve the model, which includes four operators, i. e., evaluation of the time freedom degree of each task, matching between tasks and resources, task insertion and resource update. At last, extensive experimental simulation demonstrates the effectiveness of the proposed algorithm.

**Key words:** tracking and data relay satellite; scheduling; heuristic algorithm; optimization

(上接第 10 页)

[78] WU Q, CHEN X, DING L, et al. Classification of EMG signals by BFA-optimized GSVCM for diagnosis of fatigue status[J]. IEEE transactions on automation science & engineering, 2017(99):1 – 16.

[79] LIANG J J, PAN Q K, CHEN T, et al. Solving the blocking flow shop scheduling problem by a dynamic multi-swarm particle swarm optimizer[J]. International journal of advanced manufacturing technology, 2011, 55(5/8):755 – 762.

[80] CHEN H N, ZHU Y L, HU K Y. Discrete and continuous optimization based on multi-swarm coevolution [J]. Natural computing, 2010, 9(3):659 – 682.

A Review of Bacterial Optimization and Its Applications

YAN Xiaohui<sup>1</sup>, ZHU Yunlong<sup>2</sup>, ZHANG Zhicong<sup>1</sup>, LÜ Cixing<sup>2</sup>, LI Shuai<sup>1</sup>, YI Wenjie<sup>3</sup>

(1. School of Mechanical Engineering, Dongguan University of Technology, Dongguan 523808, China; 2. School of Electrical Engineering and Intelligentization, Dongguan University of Technology, Dongguan 523808, China; 3. College of Management, Shenzhen University, Shenzhen 518060, China)

**Abstract:** Bacterial optimization was a kind of swarm intelligence approach proposed in recent year. In this paper, troduded several typical bacterial optimization algorithms were introduced and the keywords of relevant literatures were analyzed. Based on the frequency of these keywords, the algorithm improvement studies were reviewed. It manly containeded four main aspects: parameter and structure improvement, algorithm hybrid, operator improvement, and multi-objective reconstruction. Applications of bacterial optimization on engineering problems were reviewed as well. At last, the research direction in future was prospected.

**Key words:** swarm intelligence; optimization computation; bacterial optimization; algorithm improvement