

DOI:10.16356/j.1005-2615.2024.06.014

基于星间链路的天基遥感中继联合调度方法

杨志玺¹, 伍国华², 叶滢华¹, 李献斌³, 刘思力¹, 杨俊¹

(1. 国防科技大学第六十三研究所, 南京 210007; 2. 中南大学自动化学院, 长沙 410070;

3. 国防科技大学智能科学学院, 长沙 410073)

摘要: 通过星间链路 (Inter satellite link, ISL) 实现遥感卫星和中继卫星互联互通, 能将天基遥感数据及时落地, 缩短遥感反应时间, 故对卫星遥感与中继联合调度问题进行研究。在对上述问题进行描述和分析的基础上, 以最大化总的任务优先级为目标函数, 以可见时间窗、服务时间窗口以及唯一性等为约束条件构建遥感与中继联合调度规划模型。本文设计了基于自适应大规模邻域搜索框架的联合调度优化算法 (Adaptive large-scale neighborhood search based joint scheduling algorithm II, ALNS-JS-II)。该算法利用任务分配算子将多星调度分解成多个单星并行调度子问题, 利用自适应的方法对所有算子进行选择, 整个算法实现了遥感调度和中继调度间有效交互。为了验证 ALNS-JS-II 算法的有效性, 与遥感中继高耦合的自适应大规模邻域搜索算法 (Adaptive large neighborhood search based highly coupled joint scheduling algorithm I, ALNS-JS-I) 及基于自适应大规模邻域搜索的遥感中继分两阶段独立调度算法 (Two stage ALNS, Ts-ALNS) 等在多种任务场景下进行对比实验。实验结果表明, ALNS-JS-II 算法相比 ALNS-JS-I 算法以及 Ts-ALNS 算法在算法求解收益上分别提高 4.58% 和 1.48%, 在求解效率上增加 20%~30%。ALNS-JS-II 算法对遥感与中继资源联合调度问题有更好的求解能力。

关键词: 遥感卫星; 中继卫星; 星间链路; 联合调度; 自适应大规模邻域搜索

中图分类号: V19 **文献标志码:** A **文章编号:** 1005-2615(2024)06-1104-10

A Joint Scheduling Method of Space Based Remote Sensing and Relay Through Inter Satellite Link

YANG Zhixi¹, WU Guohua², YE Ganhua¹, LI Xianbin³, LIU Sili¹, YANG Jun¹

(1. The 63rd Research Institute, National University of Defense Technology, Nanjing 210007, China;

2. College of Automation, Central South University, Changsha 410070, China;

3. College of Intelligent Science, National University of Defense Technology, Changsha 410073, China)

Abstract: The interconnection between remote sensing satellites and relay satellites can be achieved through inter satellite links (ISL), which enables timely transmission of space based remote sensing data to the ground and shorten the response time of remote sensing. Thus, the joint scheduling problem of satellite remote sensing and relay is studied. On the basis of describing and analyzing the problem, taking maximizing the total task priority as the objective function, this paper constructs a remote sensing and relay joint scheduling model with constraints of visible time window, service time window and uniqueness. To solve this problem, a joint scheduling optimization algorithm based on the adaptive large-scale neighborhood search framework, the adaptive large-scale neighborhood search based joint scheduling algorithm II (ALNS-JS-II) is designed. This algorithm uses the task allocation operator to decompose the multi-satellite scheduling into

收稿日期: 2024-07-15; **修订日期:** 2024-11-21

通信作者: 伍国华, 男, 教授, 博士生导师, E-mail: guohuawu@csu.edu.cn。

引用格式: 杨志玺, 伍国华, 叶滢华, 等. 基于星间链路的天基遥感中继联合调度方法[J]. 南京航空航天大学学报, 2024, 56(6): 1104-1113. YANG Zhixi, WU Guohua, YE Ganhua, et al. A joint scheduling method of space based remote sensing and relay through inter satellite link[J]. Journal of Nanjing University of Aeronautics & Astronautics, 2024, 56(6): 1104-1113.

multiple single-satellite parallel scheduling sub-problems, and uses an adaptive method to select all operators. The whole algorithm realizes the effective interaction between remote sensing scheduling and relay scheduling. In order to verify the effectiveness of the ALNS-JS-II algorithm, it is compared with the adaptive large neighborhood search algorithm with high coupling of remote sensing and relay, the adaptive large neighborhood search based highly coupled joint scheduling algorithm I (ALNS-JS-I), the two-stage independent scheduling algorithm of remote sensing and relay based on the adaptive large neighborhood search, the two stage ALNS (Ts-ALNS), in various task scenarios. The experimental results show that the ALNS-JS-II algorithm is 4.58% and 1.48% better than ALNS-JS-I and Ts-ALNS, respectively, in terms of algorithm solution gain, and 20%—30% faster in terms of solving efficiency. In conclusion, the ALNS-JS-II algorithm has a better solving ability for the joint scheduling problem of remote sensing and relay resources.

Key words: remote sensing satellites; relay satellites; inter satellite link (ISL); joint scheduling; adaptive large-scale neighborhood search

利用遥感卫星可以全天候、全方位实时采集信息,对地面设施运行、自然灾害和交通流量等进行监测。我国天基系统境外地面站资源受限,造成天基服务响应不及时,可借助星间链路技术联通遥感和中继星,及时落地遥感数据,缩短服务时间。在利用卫星执行遥感的过程中,卫星任务可分为信息获取和信息传输两部分。在卫星执行遥感任务时,所需执行的任务量大,而可利用的卫星资源却有限,任务间存在冲突,存在无法及时执行的可能。如何在卫星任务需求与管控要求下,对卫星任务和卫星资源进行建模,消解执行遥感任务中遇到的冲突约束,最大程度地实现遥感任务收益,生成优良且稳定的调度方案是学界研究的热点^[1]。

遥感卫星调度主要有车辆路径模型^[2]、车间调度模型^[3]以及基于可见时间窗(Visible time window, VTW)的资源分配模型^[4]。中继卫星资源调度问题和遥感卫星调度模型有诸多类似^[5-8]。模型求解算法主要有启发式算法^[9-10]、元启发式算法^[11-13]以及机器学习算法^[14-16],但针对遥感与中继卫星联合调度的研究还较少。在实际应用中,观测资源与中继资源分别调度无法统筹考虑整体卫星资源,联合调度两个子问题能够更高效地利用卫星资源,完成遥感观测任务。

因此,本文以天基卫星遥感和中继联合调度为研究对象,建立遥感与中继资源联合调度的混合整数规划模型。使用基于自适应大规模邻域搜索框架的智能优化算法对模型问题进行求解,并利用改进初始解、局部搜索时引入并行计算等方法提高算法的求解质量。

1 遥感中继联合调度问题与模型

1.1 问题描述

遥感与中继联合调度过程如图1所示,整个任

务规划分为遥感卫星任务规划以及中继任务规划。

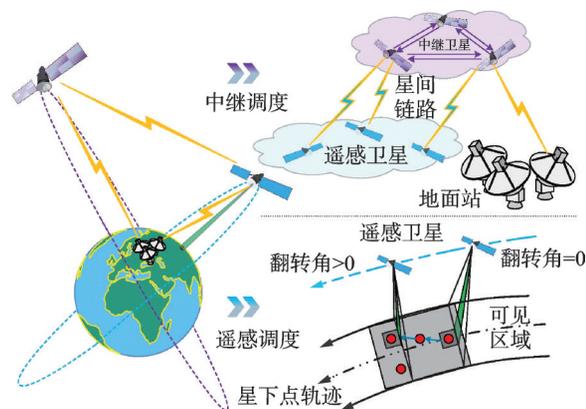


图1 遥感与中继联合调度过程

Fig.1 Process of joint scheduling of remote sensing and relay

整个卫星业务系统由3个部分组成:位于地球同步轨道的中继卫星网络层,位于中低轨道的遥感卫星服务层,以及位于地面的各种工作站和卫星管理中心。系统运作过程中,由各地用户向卫星管理中心申请观测任务,卫星管理中心根据任务的特征建立任务数据库,根据卫星运行和工作状态生成可用资源,基于此向系统发布遥感以及中继传输方案,卫星接收并执行任务,数据传回地面后,通过地面网络分发给用户。本文研究目的是在已知任务数据、卫星资源数据的基础上给出优质的任务执行方案,提高任务执行效率。

1.2 模型假设与符号说明

1.2.1 模型假设

遥感与中继联合调度问题复杂,包含约束较多,首先针对问题做出合理的假设,适当简化模型:

(1)卫星机载太阳能电池板能够为卫星活动提供足够的电力;

(2)所有用户申请的任务均是小任务或者小范围任务,能够通过1次遥感观测完成;

(3)1颗遥感卫星1次只能够执行1个任务;

(4)1颗中继卫星在同一时间仅服务1颗遥感卫星,通过星间链路中天系统维持链接;

(5)中继卫星总是与地面链接;

(6)1个地面站在可见服务时间窗内只服务1颗卫星。

1.2.2 符号及集合定义说明

遥感与中继联合调度问题的输入包括用户的观测任务申请以及可用任务资源。整个任务链包括任务分配、卫星观测和数据回传3个步骤。这些步骤一个接着一个,因此必须考虑所有类型的可用任务资源,包括遥感卫星、中继卫星和地面站。基于以上的问题设定以及分析,以下将联合调度模型中涉及的相关符号以及集合统一进行定义。

定义 $T = \{t_1, t_2, \dots, t_{|T|}\}$ 为用户申请的观测任务集。对于每一个观测任务请求 $t_i \in T$, 定义以下属性: g_i 为任务优先级, 用来衡量完成任务获得的收益; c_i 为用户最低图像质量要求, 它定义了能够满足用户要求的最低图像质量; d_i 为任务 t_i 所需观测持续时间, 以保证成像的完整性; t_{d_i} 为任务图像的传输时间, 它取决于任务图像的大小; $start_i$ 为任务最早执行时间, end_i 为任务最晚完成时间, $[start_i, end_i]$ 为任务 t_i 的服务时间窗口。

定义 $S = \{s_1, s_2, \dots, s_{|S|}\}$ 为可用遥感卫星资源集。对于每一个遥感卫星 $s_j \in S$, 其各项能力指标定义如下: M_j 为卫星 s_j 执行观测任务的最大总时间; a_1, a_2, a_3 和 a_4 分别4种不同的过渡角速度; γ, π 以及 ϕ 分别定义了卫星滚装、俯仰和航偏的最大角度。 $W_{ij} = \{w_{ij}^1, w_{ij}^2, \dots, w_{|W_{ij}|}\}$ 为观测任务 t_i 和卫星 s_j 的可见时间窗集, 其中 w_{ij}^k 为任务 t_i 和卫星 s_j 的第 k 个时间窗。对于时间窗 w_{ij}^k , 有以下的定义: b_{ij}^k 和 e_{ij}^k 分别为时间窗 w_{ij}^k 的开始时间和结束时间。 w_{ij}^{k*} 为获得最佳图像质量的时间, 通常是可见时间窗 w_{ij}^k 的中点时间。 $R = \{r_1, r_2, \dots, r_{|R|}\}$ 为可用的中继卫星集, 中继卫星与遥感卫星的可见时间窗定义为 $RW_{jr} = \{rw_{jr}^1, rw_{jr}^2, \dots, rw_{|RW_{jr}|}\}$, b_{jr}^l 和 e_{ij}^k 分别为时间窗 rw_{jr}^l 的开始和结束时间。 $G = \{g_1, g_2, \dots, g_{|G|}\}$ 为地面工作站集, 地面工作站与遥感卫星的可见时间窗定义为 $GW_{ig} = \{gw_{ig}^1, gw_{ig}^2, \dots, gw_{|GW_{ig}|}\}$, b_{ig}^m 和 e_{ig}^m 分别为时间窗 gw_{ig}^m 的开始时间和结束时间。

1.2.3 变量定义

以下针对遥感及中继联合调度模型中涉及的变量进行定义。

参考文献[17]提到的转换时间计算方法, 本文

遥感卫星执行两个相邻观测任务观测角度调整时间 $z_{ii'}$, 基于以下分段线性函数计算

$$z_{ii'} = \begin{cases} 10 + \theta_{ii'}/a_1 & \theta_{ii'} \leq 15 \\ 15 + \theta_{ii'}/a_2 & 15 < \theta_{ii'} \leq 40 \\ 20 + \theta_{ii'}/a_3 & 40 < \theta_{ii'} \leq 90 \\ 25 + \theta_{ii'}/a_4 & \theta_{ii'} > 90 \end{cases} \quad (1)$$

根据文献[17-19]的设定, $a_1 = 1.5$ ($^\circ$)/s, $a_2 = 2$ ($^\circ$)/s, $a_3 = 2.5$ ($^\circ$)/s, $a_4 = 3$ ($^\circ$)/s。

1.3 遥感与中继联合调度优化模型

式(2~20)描述了遥感与中继联合调度优化模型的目标和约束, 变量说明如表1所示。

表1 变量说明

Table 1 Variable declaration

变量	变量描述
x_{ij}^k	$x_{ij}^k \in \{0, 1\}$, 当且仅当任务 t_i 的观测时间窗位于 w_{ij}^k 内时, 取值为1
x_{ijr}^l	$x_{ijr}^l \in \{0, 1\}$, 当且仅当任务 t_i 利用时间窗 rw_{jr}^l 传输数据时, 取值为1
x_{ijg}^m	$x_{ijg}^m \in \{0, 1\}$, 当且仅当任务 t_i 利用时间窗 gw_{ig}^m 传输数据时, 取值为1
y_{ij}	$y_{ij} \in \{0, 1\}$, 当且仅当任务 t_i 被遥感卫星 s_j 执行时, 取值为1
u_i	任务 t_i 的遥感观测时间窗口起始时间戳
v_i	任务 t_i 的遥感观测时间窗口结束时间戳
u_{t_i}	任务 t_i 的中继时间窗口起始时间戳
v_{t_i}	任务 t_i 的中继时间窗口结束时间戳
$p_{ii'}$	$p_{ii'} \in \{0, 1\}$, 当且仅当任务 t_i 紧接任务 $t_{i'}$ 在同一颗遥感卫星上被观测, 取值为1
$p_{t_i t_{i'}}$	$p_{t_i t_{i'}} \in \{0, 1\}$, 当且仅当任务 t_i 紧接任务 $t_{i'}$ 在同一颗中继卫星上被传输, 取值为1
$\theta_{ii'}$	相邻两个观测任务 t_i 与 $t_{i'}$ 所需要转换的总角度
$z_{ii'}$	遥感卫星执行两个相邻观测任务 $t_i, t_{i'}$ 之间的观测角度调整过渡时间
t_{adjust}	中继卫星执行任务前的星间链路天线间对准时间
t_{rec}	结束中继后星间链路天线复位时间
$start_i$	任务 t_i 的服务时间窗口起始时间
end_i	任务 t_i 的服务时间窗口结束时间

$$\text{Max} \left(\sum_{i=1}^{|T|} \sum_{j=1}^{|S|} \sum_{l=1}^{|R|} \sum_{k=1}^{|RW_{jl}|} g_i x_{ijr}^l + \sum_{i=1}^{|T|} \sum_{g=1}^{|G|} \sum_{m=1}^{|GW_{gl}|} g_i x_{ijg}^m \right) \quad (2)$$

$$\text{s.t.} \quad \sum_{j=1}^{|S|} \sum_{k=1}^{|W_{ij}|} x_{ij}^k \leq 1 \quad t_i \in T \quad (3)$$

$$\sum_{j=1}^{|S|} \sum_{r=1}^{|R|} \sum_{l=1}^{|RW_{jr}|} x_{ijr}^l + \sum_{j=1}^{|S|} \sum_{g=1}^{|G|} \sum_{m=1}^{|GW_{gl}|} x_{ijg}^m \leq 1 \quad t_i \in T \quad (4)$$

$$b_{ij}^k \leq u_i \leq v_i \leq e_{ij}^k \quad x_{ij}^k = 1 \quad (5)$$

$$b_{jr}^l \leq u_{t_i} \leq v_{t_i} \leq e_{jr}^l \quad x_{ijr}^l = 1 \quad (6)$$

$$b_{ig}^m \leq u_{t_i} \leq v_{t_i} \leq e_{ig}^m \quad x_{ijg}^m = 1 \quad (7)$$

$$start_i \leq u_i \leq v_i \leq end_i \quad t_i \in T \quad (8)$$

$$start_{t_i} \leq u_{t_i} \leq v_{t_i} \leq end_{t_i} \quad t_i \in T \quad (9)$$

$$v_i + z_{i'} \leq u_{i'} \quad p_{i'} = 1 \quad (10)$$

$$v_i + t_{\text{adjust}} + t_{\text{rec}} \leq u_{i'} \quad p_{i'} = 1 \quad (11)$$

$$v_i = u_i + d_i \quad (12)$$

$$v_i = u_i + t_d \quad (13)$$

$$\sum_{i=1}^{|T|} y_{ij} d_i \leq M_j \quad s_j \in S \quad (14)$$

$$y_{ij} \in \{0, 1\} \quad s_j \in S, i = 1, 2, \dots, |T| \quad (15)$$

$$x_{ij}^k \in \{0, 1\} \quad t_i \in T, \omega_{ij}^k \in W_{ij}, s_j \in S \quad (16)$$

$$x_{ijr}^l \in \{0, 1\} \quad t_i \in T, \text{rw}_{jr}^l \in \text{RW}_{jr}, s_j \in S, r_l \in R \quad (17)$$

$$x_{ijg}^m \in \{0, 1\} \quad \text{gw}_{jg}^m \in \text{GW}_{jg}, s_j \in S, g_m \in G, t_i \in T \quad (18)$$

$$p_{i'} \in \{0, 1\} \quad p_{i'} \in \{0, 1\}, t_i, t_i' \in T \quad (19)$$

$$\theta_i \geq 0, z_i \geq 0 \quad t_i \in T \quad (20)$$

式(2)表示使所有完成遥感观测以及数据传输的任务总的优先级最大;式(3)以及式(4)是唯一性约束,式(3)表示任务只能被观测1次,式(4)代表任务只能被传输1次;式(5~7)为可见时间窗约束,限制1个任务的观测以及中继时间窗必须包含在1个可见时间窗内;式(8~9)为服务时间窗口约束,限制1个任务的观测时间以及中继时间窗必须包含在其服务时间窗口内;式(10~11)为操纵约束,式(10)表示两个相邻的观测行为之间需要间隔一定的时间以便卫星调整其观测角度,式(11)表示通过中继卫星进行传输两个相邻的任务需要间隔一定的时间以便调整其天线;式(12)以及式(13)保证任务有足够的时间窗口;式(14)为内存约束,它限制每个遥感卫星的观测总时间为 M_j ;式(15~20)定义了变量的取值范围。本优化模型是一个复杂的组合优化问题,涉及复杂的约束条件,求解困难。

2 基于自适应大规模邻域搜索的遥感中继联合调度算法

基于自适应大规模邻域搜索的遥感中继联合调度算法(Adaptive large-scale neighborhood search based joint scheduling algorithm II, ALNS-JS-II)设计为3层结构,旨在优化遥感卫星与中继卫星的联合调度。第1层是自适应任务分配层,设计了多种任务调度算子来将复杂的多星调度问题分解为更易于处理的单星调度子问题。第2层专注于单颗遥感卫星的并行调度,其中集成了多种调度算子来优化每个卫星的任务分配。如果在一定迭代次数内这些子问题的总回报没有显著改善,算法将自适应地调用第1层,重新分配部分任务给不同的卫星,同时根据性能反馈调整算子权重。第3层是自适应中继调度层,在这一层中,同样设计了多种算子来规划中继任务,并根据算子的性能反馈

动态调整其权重。同时,为了提高各阶段算子的适应性,设计了自适应算子权重调整机制。该机制依据算子历史表现来动态调整其权重,并影响其被选择的频率。算子得分基于其生成解的质量,得分高则权重增加,更易被选中。最后,整个搜索过程将持续进行,直至满足终止条件。

算法1 基于自适应大规模邻域搜索的遥感中继联合调度算法

输入: $T, S, R, G, W_{ij}, \text{GW}_{jg}, \text{RW}_{jr}$,

$i \in \{1, 2, \dots, |T|\}$,

$j \in \{1, 2, \dots, |S|\}, r \in \{1, 2, \dots, |R|\}$,

$g \in \{1, 2, \dots, |G|\}$

输出: S_B, SD_B //最优调度方案

(1) 将任务按照优先级排序,使用贪婪搜索生成初

始解 S_0, SD_0 ;计算初始解的收益 r_0, R_0 ;

(2) for each $s_i \in S$ do

(3) $S_C^s \leftarrow S_0^s, S_B^s \leftarrow S_0^s, r_C^s \leftarrow S_0^s, r_B^s \leftarrow S_0^s$,

$\text{SD}_C \leftarrow \text{SD}_0, \text{SD}_B \leftarrow \text{SD}_0, R_C \leftarrow R_0$,

$R_B \leftarrow R_0$;

(4) end for

(5) while 未达到最大迭代次数或任务未全调度 do

(6) for each $s_i \in S$ do

(7) 选择1个删除算子,从方案 S_C^s 中删除 n 个任务,得到摧毁方案 S_D^s ;

(8) 所有未安排的任务构成集合 Q^s ;

(9) end for

(10) if $\eta = \text{true}$ then

(11) $\eta \leftarrow \text{false}$,将所有 Q^s 构成集合 Q ;

(12) 选择一个任务分配算子,将集合 Q 里面的任务重新分配给卫星得到新的集合 Q^s ;

(13) if 每 ρ 次任务分配结束 do

(14) 更新任务分配算子权重,并初始化算子得分;

(15) end if

(16) end if

(17) for 每颗遥感卫星的摧毁方案 S_D^s do

(18) 选择一个插入算子,将集合 Q^s 中的任务插入方案 S_D^s ;

(19) 得到新的遥感观测方案 S_N^s ;计算方案收益 r_N^s ;

(20) if $r_N^s > r_C^s$ then

(21) $S_C^s \leftarrow S_N^s, S_B^s \leftarrow S_N^s, r_C^s \leftarrow r_N^s, r_B^s \leftarrow r_N^s$;

(22) else

(23) 使用模拟退火准则判断是否接受新的方案 S_N^s ;

- (24) end if
 (25) 更新选择的删除算子和插入算子的得分;
 (26) if 迭代次数整除 φ 时 then
 (27) 更新所有遥感启发式算子的权重;初始化算子得分;
 (28) end if
 (29) end for
 (30) 将所有遥感卫星的观测任务方案 S_C^s 汇总组成总的观测方案 S_C ;
 (31) 选择一个启发式中继任务分配算子;
 (32) for 遥感方案 S_C 中所有的任务 do
 (33) 使用选定的启发式算子将任务 t_i 进行中继分配;
 (34) end for
 (35) 得到中继方案 SD_C , 计算其收益 R_C ;
 (36) if $R_C > R_B$ then
 (37) $S_B \leftarrow S_C, SD_B \leftarrow SD_C$
 (38) end if
 (39) 更新算子得分
 (40) if 每 γ 次迭代结束 then
 (41) 更新所有中继任务算子的权重;然后初始化算子得分;
 (42) end if
 (43) if 总的收益经过 μ 次迭代后均没有改进 then
 (44) $\eta = \text{true}$;
 (45) end if
 (46) end while
 (47) return S_B, SD_B

2.1 初始解生成

算法初始解用贪婪思想生成。首先分别使用4个任务分配算子对任务进行分配,每个任务分配算子分配任务后,生成一组并行的单卫星调度问题。对每一组中的单个卫星调度问题,将任务按任务优先级的非递增顺序排序,如果两个任务具有相同的优先级,则选择任务VTW开始时间较早的任务优先排序。根据此顺序,依次对每个任务确定观测时间,如此每个任务分配算子都将产生一个可行的临时遥感观测方案,针对每个遥感观测方案,利用贪婪搜索思想继续生成中继方案,对每个中继方案,计算其收益 R ,选择收益最大的一个方案作为初始解 S_0 。

2.2 遥感任务调度算子设计

2.2.1 任务分配算子设计

当所有子问题(单颗卫星)解决方案的总回报

在连续 μ 代没有改善时,将会解锁自适应任务分配层。任务分配算子集合包含4个启发式算子,它们分别依据不同的规则将任务分配给卫星。

算子1:随机分配算子。这是最简单的分配算子,通过该算子,可以随机地将任务分配给不同的卫星。

算子2:冲突分配算子。任务VTW之间重叠时间越长,表明在确定每个任务的观测时间窗(Observation window, OW)时将会有更多的冲突。任务分配给冲突程度最低的卫星有利于局部搜索。可见时间窗 w_{ij}^k 的冲突度用 d_{ijk}^c 表示,计算公式为

$$d_{ijk}^c = \frac{\sum_{w \in \text{Over}(w_{ij}^k)} \text{TimeSpan}(w_{ij}^k, w)}{e_{ij}^k - b_{ij}^k} \quad (21)$$

式中 $\text{Over}(w_{ij}^k)$ 表示与 w_{ij}^k 重叠的一组可见时间窗。TimeSpan函数用于计算重叠时间长度。

卫星 s_j 上的任务 t_i 的冲突度计算公式为

$$d_{ij}^c = \frac{\sum_{k=1}^{|W_{ij}|} d_{ijk}^c}{|W_{ij}|} \quad (22)$$

算子3:机会分配。VTW越多说明任务被成功调度的机会越大。因此,将任务分配给机会度最大的卫星有利于收益极大化。机会度通过汇总所有VTW的持续时间来确定,计算公式为

$$d_{ij}^o = \sum_{k=1}^{|W_{ij}|} (e_{ij}^k - b_{ij}^k) \quad (23)$$

算子4:经验分配。顾名思义,该算子通过历史经验来选择任务卫星。更精确地说该算子通过根据最近几次迭代方案的收益(目标函数值)来分配任务。令 d_e 为经验度,并令 $R = \{r_1, r_2, \dots, r_{|R|}\}$ 为任务 t_i 分配给卫星 s_j 的收益。经验度定义为历史收益的平均水平,计算公式为

$$d_{ij}^e = \frac{\sum_{i=1}^{|R|} r_i}{|R|} \quad (24)$$

在最初几次迭代时,由于没有历史经验,为了鼓励任务尝试新的卫星,将经验度初始化为一个中等偏上的值。

2.2.2 遥感调度邻域算子设计

根据算法结构,通过反复调用一系列破坏算子和修复算子,搜索解空间。这些算子提供新的解搜索途径,以下对破坏算子和修复算子进行定义。

(1) 破坏算子

本部分定义了4个删除算子。这些算子用于从方案中删除一些任务,以便插入新的任务形成新的方案。每个删除算子从解序列中删除 n 个任务,

移除它们,并将它们保存在任务库 Q 当中,得到销毁的解决方案 S_D^s 。

算子1:随机删除算子。此算子为简单的启发式算子,从给定的任务序列中随机删除 n 个任务。

算子2:优先级删除算子。此算子是一个贪婪启发式算子。从任务序列中选择并删除 n 个优先级最低的任务。

算子3:机会删除算子。卫星可能存在多个 VTW 来观测计划内的任务。首先计算每个任务的 VTW 数目。删除任务时,优先删除具有更多窗口的任务,因为相比于任务时间窗口较少的任务此类任务被观测的可能性更大。这样可以使得任务灵活性较低的任务得到保留。

算子4:冲突删除算子。冲突程度 cd^* 表示 VTW 与其他时间窗口重叠程度。计算公式为

$$cd^* = \frac{\sum_{tw_{ij} \in \text{Over}(tw^*)} \text{TimeSpan}(tw^*, tw_{ij})}{\text{Over}(tw^*)} \quad (25)$$

$tw_{ij} \in F_r \cup Q$

式中: tw^* 为当前时间窗口; tw_{ij} 为非计划时间窗口; $\text{Over}(tw^*)$ 为与 tw^* 重叠的时间窗口集; TimeSpan 函数用来计算重叠的长度,如果 $\text{TimeSpan}(tw^*, tw_{ij}) > 0$, 则 tw^* 与 tw_{ij} 重叠; F_r 表示没有计划的非计划任务集。冲突程度是当前时间窗口与非计划时间窗口冲突的平均时间。因为在插入新的任务时,冲突程度大的时间窗口更有可能造成冲突,所以算子在运行中应优先删除冲突程度大的时间窗口。

(2) 修复算子

插入算子通过向已经销毁的解决方案 S_D^s 中插入新的任务来获得新的可行方案 S_N^s , 以求获得更优的解。待插入的任务由两部分组成:一个是没有被安排的任务;另一个是从当前解中删除的任务。这一部分设计了3个不同的算子。

算子1:贪婪插入算子。该算子通过对所有候选任务根据任务优先级进行排序,然后按照顺序依次插入任务。在插入过程中,依次遍历任务所有时间窗口,直到任务调度成功,否则,任务调度失败。

算子2:最小机会插入。该算子优先安排观测时间窗口较少的任务。随着算法的发展,任务插入的可能性会变小,因此优先安排时间窗口较少的任务可以尽可能成功调度更多的任务。该算子与删除算子3类似,不过本算子的机会是指 VTW 数量。

算子3:最小冲突插入。由于该算法的目的是构建一个没有冲突的任务序列,故本算子用来插入

一个任务,该任务将最小化与其他未插入任务的冲突。该部分冲突度的定义与删除算子4类似。

2.3 中继调度算子设计

在完成遥感调度,生成遥感调度任务方案 S_C 之后,通过本部分的相关机制分配中继资源,产生中继调度方案,本部分一共设计3个启发式算子来调度中继资源。

算子1:优先级传输算子。该算子通过针对遥感调度方案 S_D^s 中的任务按照任务优先级进行排序,然后按照优先级顺序依次将任务插入解序列,并擦除被任务占据的时间窗口资源。直到无法插入任务时,得到中继调度方案 SD_C 。

算子2:最小冲突度传输算子。该算子首先针对所有的任务进行冲突度计算,冲突度计算方式与式(25)类似,但在计算时 tw^* 只考虑任务遥感观测任务时间点之后的时间窗口。然后依据冲突度排序,优先调度冲突度小的任务。

算子3:贪婪传输算子。该算子的传输策略是单位时间完成的任务收益最大化,在运行过程中,该算子首先计算中继任务的单位权重为

$$\bar{w}_i = g_i / t_d \quad (26)$$

得到各个任务的单位时间中继收益之后,将任务进行排序,优先插入单位收益 \bar{w}_i 更大的任务。

2.4 自适应调整算子权重机制设计

为提升算法的效率和适应性,本文为每个算子分配了得分和权重,这两个参数共同决定了算子在算法中的选择概率。算子的得分反映了其历史表现,即它在以往迭代中生成的可行方案的收益。得分越高,表明算子在以往的任务中越有效,因此其权重也会相应增加。这种设计使得算法能够根据算子的实际表现动态调整其在未来迭代中的重要性。

在自适应任务分配层、自适应遥感任务调度层以及自适应中继任务调度层,都实施了这一自适应机制。以遥感任务调度为例,每个算子在两次权重调整之间会经历多个算法迭代片段。在这些迭代中,每当算子生成一个新的解决方案,根据该方案的收益对其进行评分。算子的权重会根据其历史累计评分进行调整,评分越高,算子的权重增加得越多,反之亦然。完成权重调整后,重置算子的评分,以便在下一个迭代片段中重新开始评分过程。

这种自适应调整算子权重的机制使得算法能够不断学习并优化其搜索策略,从而在复杂的调度问题中找到更优的解决方案。通过这种方式,算法能够自适应地调整算子的使用频率,优先选择在过去表现优异的算子,同时减少效果不佳算子的使

用,以提高整体的调度效率和效果。针对每个算子设置两个参数:得分以及权重。分别以 w_i^d 和 π_i^d 表示摧毁算子的权重和分数,以 w_i^r 和 π_i^r 分别表示修复算子的权重和分数。分数用来评价算子在求解优质方案的有效性,权重用来指示算子的重要性,权重根据算子分数进行更新,分数更新规则如下。

(1) $\pi_i = \pi_i + \sigma_1$: 如果得到了一个新的全局最优解;

(2) $\pi_i = \pi_i + \sigma_2$: 如果新的解不是全局最优解但比当前解要好;

(3) $\pi_i = \pi_i + \sigma_3$: 如果算子获取了一个比当前解更差的解,但根据模拟退火准则被接受;

(4) $\pi_i = \pi_i + \sigma_4$: 如果新的解被舍弃。

π_i 代表选定算子的评分, $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ 分别为不同的增加分数。

这些规则通过接受一些非改进性的解,实现了搜索的多样性。每隔一定的迭代次数,根据式(27)更新算子的权重

$$\lambda = (1 - \lambda)w_i + \lambda\pi_i / \sum_{i=1}^H \pi_i \quad (27)$$

式中: $\lambda \in [0, 1]$,它控制算子权重受算子历史表现的敏感性,为0表示权重保持不变,为1表示历史性能没有影响,仅仅取决于算子的分数; H 表示算子的数量。

在选择算子时采取的方法时轮盘赌机制。算子 i 被选择的概率为

$$P_i = w_i / \sum_{j=1}^H w_j \quad (28)$$

2.5 遥感调度方案接受准则

对于产生的新的解序列和原解序列,采取模拟退火准则(Simulated annealing, SA)决定是否接受非改进性方案,设 τ 为温度参数。如果一个新的解的收益 R_N 小于当前收益 R_C ,则新的解被接受的概率为

$$P = \exp\left(\frac{100}{\tau} \left(\frac{R_N - R_C}{R_C}\right)\right) \quad (29)$$

在每次迭代中,温度 τ 线性变化,设 τ_n 为第 n 次迭代时的温度,则 $\tau_n = c\tau_{n-1}$, c 为退火系数。为了保证退火过程缓慢且平坦,初始温度设置为

$$\tau_0 = \frac{-0.05}{\ln 0.5} \times r_0 \quad (30)$$

当满足:(1)达到了最大迭代次数;(2)所有有时间窗口的任务都被安排完毕,迭代过程停止。

2.6 任务插空策略

在进行任务插入的时候,有3种常用的任务插空策略:紧前插入策略、随机插入策略和紧后插入策略。3种插空策略如图2所示,本文使用紧前插空策略。

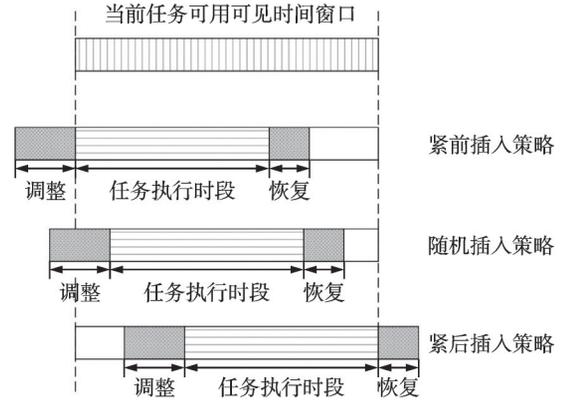


图2 任务插空策略

Fig.2 Strategy of task insertion

3 仿真实验

为了评估本文提出的算法性能,本文通过建立不同规模任务规划场景,将提出算法与两种基于自适应大规模搜索框架的算法进行比较,并测试其在不同场景下的性能。

3.1 实验场景设计

为了对本文提出的针对遥感中继资源联合调度的算法进行测试,构建实验仿真场景,场景历时24 h,以下分别对场景任务构建和卫星设置进行阐述。

3.1.1 任务需求仿真

任务在经度 $0^\circ\text{E} \sim 120^\circ\text{E}$,纬度 $3^\circ\text{N} \sim 50^\circ\text{N}$ 内随机生成观测目标。对于每个任务,包含以下因素:优先级、地理位置和观测持续时间。任务优先级是一个 $[0, 10]$ 范围内的随机整数。设置任务获取图片数据量大小在 $10 \sim 120$ Mb之间,数据传输速率为 100 kb/s。

3.1.2 实验场景参数设置

实验总共设置了6颗遥感卫星,其初始轨道参数如下:半长轴 a 为 $7\,200$ km,偏心率 e 为 6.27×10^{-4} ,卫星倾角为 96.576 ,近地点角度 ω 为 0 。其余参数设置如表2所示。每个遥感卫星设置1个星载传感器对地面目标进行观测。实验设置3颗中继卫星,3个地面站用于数据传输与接收。其中每个中继卫星设置一个单址天线用于执行星间链路,天线对准时间设置为 30 s,复位时间设置为 20 s。遥感卫星姿态转换时间统一定义为 20 s。

任务重新分配迭代次数: $u = 4$;算子权重更新步长: $\lambda = 6$;任务删除百分比: 10% ;算子得分增量: $\sigma_1 = 30, \sigma_2 = 20, \sigma_3 = 10, \sigma_4 = 1$;算子权重更新迭代次数: $\psi = 10, \rho = 5$;最大迭代次数: 500 ;退火系数: 0.95 。

表2 遥感卫星轨道参数

卫星号	上升节点赤经/(°)	平近点角/(°)
卫星1	175.72	0.075
卫星2	145.72	30.075
卫星3	115.72	60.075
卫星4	85.72	90.075
卫星5	55.72	120.075
卫星6	25.72	150.075

3.2 对比算法

本文在建立算法框架时,尝试了其他两种针对遥感卫星与中继资源联合调度的算法用于对比。分别为遥感中继高耦合的自适应大规模邻域搜索算法(Adaptive large-scale neighborhood search based highly coupled joint scheduling algorithm I, ALNS-JS-I)、基于自适应大规模邻域搜索的遥感中继分两阶段独立调度算法(Two-stage-ALNS, Ts-ALNS)。Ts-ALNS算法将遥感资源调度和中继资源调度视为两个独立的调度过程,分别设计了自适应大规模邻域搜索算法进行优化,它们唯一的交互在遥感调度结束后,将遥感调度方案输入到中继调度优化算法中。而ALNS-JS-I算法将遥感调度和中继调度耦合在一起协同调度。

3.2.1 遥感中继高耦合自适应大规模邻域搜索算法

ALNS-JS-I与ALNS-JS-II不同的地方在于在删除遥感调度任务的同时删除对应的中继调度任务,并在中继调度阶段删除一定的任务,最后使用中继插入算子恢复解序列,生成新的中继方案,遥感调度与中继调度耦合程度更高,同时该算法相比ALNS-JS-II也更为复杂。具体算法步骤如下:

步骤1 将观测任务分配给卫星,然后根据贪婪思想生成初始遥感调度方案、中继方案;

步骤2 选择遥感任务删除算子从遥感调度方案中删除部分任务,若该任务在中继方案中也被成功调度,在中继方案中也进行删除;

步骤3 判断是否重新进行任务分配,是则自适应选择一个遥感任务分配算子对未调度及删除的任务重新分配给卫星;

步骤4 选择一个遥感任务插入算子将任务重新插入,生成新的遥感调度方案;

步骤5 选择一个中继任务删除算子,从中继方案中删除部分任务;

步骤6 选择一个中继任务插入算子,向中继方案中插入任务,生成新的中继方案;

步骤7 计算新的遥感调度方案的收益值,使用SA准则对新的方案进行接受判断;

步骤8 对被选中的算子进行评分或权重更新;

步骤9 判断终止条件,若达到最大迭代次数或者所有的任务均被调度,则停止迭代,否则返回步骤2;

步骤10 输出联合调度方案。

3.2.2 基于自适应大规模邻域搜索的遥感中继分两阶段独立调度算法

Ts-ALNS算法是将遥感与中继资源的联合调度问题分割成两个独立的调度过程,分别使用自适应大规模邻域搜索算法求解,遥感调度输出一个满意的遥感观测方案作为中继阶段的输入参数,该算法遥感调度和中继调度交互较少,联系不紧密,算法步骤如下:

步骤1 使用自适应大规模邻域搜索算法生成一个满意的遥感调度方案;

步骤2 使用自适应大规模邻域搜索算法生成一个满意的中继调度方案;

步骤3 输出联合调度方案。

3.3 实验结果

实验在配置为Intel(R) Core(TM) i7-8550U CPU、8 GB RAM的计算机上进行,程序使用Python 3.9.6解释器实现。使用MATLAB调用STK进行资源与任务需求仿真,然后将获取的实验数据导入Jupyter Notebook开展算法测试。

运行结果如表3所示。在600个任务的联合调度场景下,ALNS-JS-II算法在运行时间和任务收益完成率(成功调度的任务的收益与对地观测任务的总收益之比)都优于其他两个算法,整体性能更高。在收益完成率上,ALNS-JS-II分别比其余算法高4.58%和1.48%。在求解效率上,ALNS-JS-II算法分别比其余算法求解效率高出20%~30%。

表3 算法运行结果

评价指标	ALNS-JS-II	ALNS-JS-I	Ts-ALNS
任务申请数量	600	600	600
总的任务收益	3 384	3 384	3 384
任务完成总收益	2 831	2 676	2 781
任务收益完成率/%	83.66	79.08	82.18
算法执行时间/s	12 430.40	16 270.54	17 664.95

3.4 算法性能测试分析

为了验证本文所提算法的性能,依次从任务规模差异、算法收敛性和遥感卫星数量差异等角度对算法进行分析。

3.4.1 任务规模差异实验

6组(E1~E6)实验结果如表4所示,实验参数配置如表5所示。

表4 不同任务规模下3种算法性能测试结果

Table 4 Performance results for three algorithms at different task scales

实验分组	运行时间/s			收益率/%		
	ALNS-JS-II	ALNS-JS-I	Ts-ALNS	ALNS-JS-II	ALNS-JS-I	Ts-ALNS
E1	156.7	98.2	212.7	100	100	100
E2	1 678.8	1 932.4	2 454.5	99.84	99.42	100
E3	4 044.5	4 866.8	5 641.3	98.60	97.21	98.34
E4	6 516.6	7 860.08	8 725.5	96.50	94.52	96.48
E5	8 653.7	12 885	1 386.5	89.69	87.74	88.52
E6	13 050.4	16 838.4	17 868.5	83.58	78.98	82.54

表5 实验参数设置

Table 5 Experimental parameter setting

实验分组	E1	E2	E3	E4	E5	E6
任务规模	100	200	300	400	500	600

实验重复运行了3次取平均值。实验结果表明,任务规模较小时,3种算法求解收益相差不大;当任务规模增大,ALNS-JS-II算法的求解收益指标逐渐比其他两种算法更优,在算法搜索效率方面,ALNS-JS-II算法在各个任务规模均比其他两种算法有较好的表现。

3.4.2 算法收敛性分析

为了研究ALNS-JS-II在不同任务规模下算法收敛性,对不同任务规模进行了对比实验。结果如图3所示,实验结果表明,在可控任务规模以内,ALNS-JS-II算法都能够在有限迭代次数内收敛并获取联合调度方案,说明该算法在遥感与中继联合调度中具有良好的适用性。

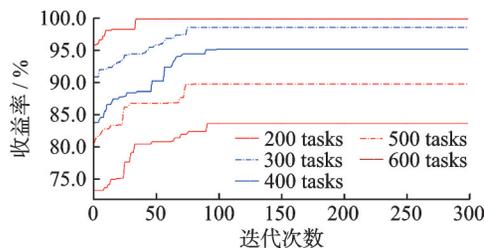


图3 ALNS-JS-II收敛曲线

Fig.3 Convergence curves of ALNS-JS-II

3.4.3 遥感卫星数量差异实验

遥感卫星是联合调度问题中重要的资源,决定着对地观测任务被观测的可能性。其数量越多,则对地观测任务之间的VTW越多,同时遥感卫星和

中继卫星的可见时间窗也越多,但问题的复杂度越大。为研究遥感卫星数量对ALNS-JS-II算法性能的影响,本文分别在不同数量遥感卫星数量下进行实验。结果如图4所示,ALNS-JS-II在不同遥感卫星数量下均具有较好的求解性能,并在任务规模变大时,逐渐优于其他两种算法。

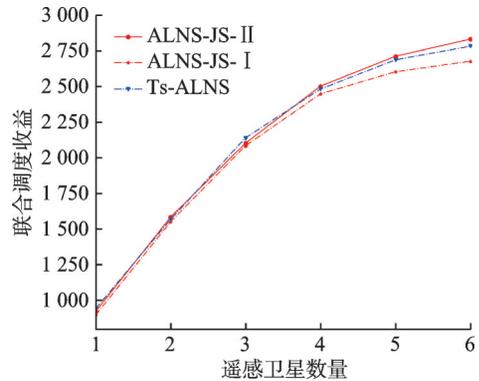


图4 不同数量遥感卫星下3种算法性能测试比较

Fig.4 Comparison of performance for three algorithms with different remote sensing satellites

4 结 论

本文主要完成的工作和得到的结论如下:

(1)采用分而治之的思想分解多颗遥感卫星任务调度问题。为了降低任务规模和任务调度难度,提高任务调度效率,本文采取使用分而治之的思想,将多颗遥感卫星任务调度分解成多个单颗遥感卫星任务调度。各个子调度问题相互隔离,互不干扰,通过任务分配进行任务交互,最终得到整体的遥感调度方案。这种方法可以缩小任务规划规模、降低任务规划难度,使得规划过程更加简洁清晰,有效降低算法难度。

(2)设计基于自适应大规模邻域搜索的联合调度算法。本文设计了以自适应大规模邻域搜索算法为基础的联合调度算法,该算法包括任务分配过程、遥感调度过程、中继调度过程。每个过程均设计了若干相应的启发式算子,这些算子通过各自的权重决定其被调用的概率,算法迭代过程对算子产生反馈,调节算子权重,提高了算法的求解能力。采用本文方法后,求解效率提升20%~30%。

参考文献:

[1] 杜永浩,邢立宁,姚锋,等.航天器任务调度模型、算法与通用求解技术综述[J].自动化学报,2021,47(12): 2715-2741.
DU Yonghao, XING Lining, YAO Feng, et al. Survey on models, algorithms and general techniques for spacecraft mission scheduling[J]. Acta Automatica Sinica, 2021, 47(12): 2715-2741.

- [2] 郭玉华,李军,赵珂,等. 多星联合任务规划中的启发式求解方法研究[J]. 宇航学报, 2009, 30(2): 652-658.
GUO Yuhua, LI Jun, ZHAO Ke, et al. A heuristic method for earth observing satellites united imaging scheduling[J]. Journal of Astronautics, 2009, 30(2): 652-658.
- [3] XIAO Y, ZHANG S, YANG P, et al. A two-stage flow-shop scheme for the multi-satellite observation and data-downlink scheduling problem considering weather uncertainties[J]. Reliability Engineering & System Safety, 2019, 188: 263-275.
- [4] MOK S H, JO S, BANG H, et al. Heuristic-based mission planning for an agile earth observation satellite [J]. International Journal of Aeronautical and Space Sciences, 2019, 20(3): 781-791.
- [5] 郭超,熊伟,郝利云. 基于双层优先级的中继卫星系统任务调度算法[J]. 计算机应用研究, 2018, 35(5): 1506-1510.
GUO Chao, XIONG Wei, HAO Liyun. Relay satellites system task scheduling algorithm based on double-layer priority[J]. Application Research of Computers, 2018, 35(5): 1506-1510.
- [6] SHE Y, LI S, ZHAO Y. Onboard mission planning for agile satellite using modified mixed-integer linear programming[J]. Aerospace Science and Technology, 2018, 72: 204-216.
- [7] 贺川,李亚晶,丘震. 按需申请模式下的中继卫星任务规划模型与算法设计[J]. 中国空间科学技术, 2017, 37(6): 46-55.
HE Chuan, LI Yajing, QIU Zhen. Task programming models and algorithms of tracking and data relay satellite in application on demand[J]. Chinese Space Science and Technology, 2017, 37(6): 46-55.
- [8] ZHAO W, ZHAO J, ZHAO S, et al. Resources scheduling for data relay satellite with microwave and optical hybrid links based on improved niche genetic algorithm[J]. Optik, 2014, 125(13): 3370-3375.
- [9] CHEN X, REINELT G, DAI G, et al. Priority-based and conflict-avoidance heuristics for multi-satellite scheduling[J]. Applied Soft Computing, 2018, 69: 177-191.
- [10] HE L, LIU X, LAPORTE G, et al. An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling[J]. Computers & Operations Research, 2018, 100: 12-25.
- [11] NIU X, TANG H, WU L. Satellite scheduling of large areal tasks for rapid response to natural disaster using a multi-objective genetic algorithm[J]. International Journal of Disaster Risk Reduction, 2018, 28: 813-825.
- [12] CHEN H, LI L, ZHONG Z, et al. Approach for earth observation satellite real-time and playback data transmission scheduling[J]. Journal of Systems Engineering and Electronics, 2015, 26(5): 982-992.
- [13] 李夏苗,陈新江,伍国华,等. 考虑断点续传的中继卫星调度模型及启发式算法[J]. 航空学报, 2019, 40(11): 274-289.
LI Xiamiao, CHEN Xinjiang, WU Guohua, et al. Scheduling model and heuristic algorithm for tracking and data relay satellite considering breakpoint transmission[J]. Acta Aeronautica et Astronautica Sinica, 2019, 40(11): 274-289.
- [14] WANG H, YANG Z, ZHOU W, et al. Online scheduling of image satellites based on neural networks and deep reinforcement learning[J]. Chinese Journal of Aeronautics, 2019, 32(4): 1011-1019.
- [15] HE Y, XING L, CHEN Y, et al. A generic Markov decision process model and reinforcement learning method for scheduling agile earth observation satellites [J]. IEEE Transaction on Systems, Man, and Cybernetics, 2020, 52(3): 1463-1474.
- [16] ZHANG J, XING L. An improved genetic algorithm for the integrated satellite imaging and data transmission scheduling problem[J]. Computer & Operation Research, 2022, 139: 105626.
- [17] LIU X, LAPORTE G, CHEN Y, et al. An adaptive large neighborhood search metaheuristic for agile satellite scheduling with time-dependent transition time[J]. Computers & Operations Research, 2017, 86: 41-53.
- [18] PENG G, DEWIL R, VERBEECK C, et al. Agile earth observation satellite scheduling: An orienteering problem with time-dependent profits and travel times [J]. Computers & Operations Research, 2019, 111: 84-98.
- [19] PENG G, SONG G, XING L, et al. An exact algorithm for agile earth observation satellite scheduling with time-dependent profits[J]. Computers & Operations Research, 2020, 120: 104946.