

基于启发式算法的卫星反应式调度

张 铭¹ 卫 波² 王晋东¹

(解放军信息工程大学 郑州 450001)¹ (北京市遥感信息研究所 北京 100101)²

摘 要 面对地震、火灾等突发性事件,需要对卫星调度方案进行动态调整。文中考虑了卫星资源失效和应急任务加入等动态不确定性因素,综合任务约束、时间约束、卫星能量和存储约束条件,设计了基于触发规则的事件驱动策略,构建了以最大化调度收益和最小化扰动测度为目标函数的反应式调度多目标优化模型,提出了基于任务迫切度的选择策略、基于时间和角度的合成策略、基于冲突程度的替换策略,最后采用了一种考虑任务合并、插入、移位、替换的启发式算法。仿真结果表明,相比事件驱动和周期驱动策略,文中所设计的基于触发规则的事件驱动策略能够兼顾触发次数、任务完成率和响应时间,是一种有效的反应式驱动策略,MISR-HA(Heuristic Algorithm for Merging, Inserting, Shifting and Replcing)算法相比其他 3 种算法在调度收益上平均提高了 14.78%,在扰动测度上平均降低了 41.91%,在运行时间上平均缩短了 14.63%,从而有效地证明了该算法的有效性。

关键词 约束条件,触发规则,调度收益,扰动测度,启发式

中图分类号 TP79 文献标识码 A DOI 10.11896/jsjcx.180901806

Satellite Reactive Scheduling Based on Heuristic Algorithm

ZHANG Ming¹ WEI Bo² WANG Jin-dong¹

(PLA Information Engineering University, Zhengzhou 450001, China)¹

(Beijing Institute of Remote Sensing Information, Beijing 100101, China)²

Abstract For the sudden events such as earthquakes and fires, it is necessary to dynamically adjust the satellite dispatching plan. Considering the dynamic uncertainty factors such as satellite resource failure and joining emergency task, combining task constraints, time constraints, satellite energy and storage constraints, this paper designed an event-driven strategy based on trigger rules, constructed a multi-objective optimization model regarding maximized scheduling gain and minimum disturbance measure as objective function, and then proposed a heuristic algorithm considering task merging, insertion, shifting and replacement. The simulation results show that the event-driven strategy based on trigger rules can balance the number of triggers, task completion rate and response time, and it is an effective reactive driving strategy. Compared with other three algorithms, the MISR-HA algorithm improves the scheduling gain by an average of 14.78%, reduces the disturbance measurement by an average of 41.91%, and reduces the running time by an average of 14.63%, thus proving the effectiveness of the algorithm.

Keywords Constraint condition, Trigger rule, Dispatch income, Disturbance measure, Heuristic method

1 引言

目前卫星调度主要以静态调度为主,即在外界信息不变的情况下,将卫星资源无冲突地分配给各个任务,最终生成调度方案,从而保证任务调度相互不冲突。但在实际应用中,外界条件一般是不确定的,卫星在太空运行时可能会受到电磁干扰等因素的影响,导致其暂时性失效。而卫星的修复需要一定的时间,需要根据实际情况将卫星对应的任务进行重新分配调度,以保证系统调度的稳定性。同时,地震、火灾等自然灾害往往发生得比较突然,需要在短时间内对其进行调度观测,对于此类调度任务应尽可能满足其对动态性和时效

性^[1]的要求。因此,研究不确定条件下卫星资源的动态重调度具有一定的现实意义。

卫星资源失效的随机性、应急任务到达的动态性和时效性、任务观测时间窗口的不定性、卫星任务调度约束的复杂性等加大了问题求解的难度,引起了国内外许多学者的兴趣。文献[2]考虑到用户的实际需求,结合任务到达的动态性和随机性特点,设计了一种基于优先级的时序前瞻搜索算法,以实现调度的高效性和系统的稳定性。但该方法没有考虑卫星资源失效的随机性以及应急任务的时效性等实际情况。文献[3]主要针对卫星资源失效的情况,基于诚信机制的招投标合同网机制,提出了一种基于 Multi-Agent 的任务重调度模型。

到稿日期:2018-09-27 返修日期:2018-12-06 本文受十三五预研项目(30503020102),军内科研项目(TJ20172A03067)资助。

张 铭(1993—),男,硕士生,主要研究方向为资源调度,E-mail:2663931027@qq.com;卫 波(1990—),男,硕士,工程师,主要研究方向为遥感卫星任务控制;王晋东(1966—),男,教授,主要研究方向为信息安全、服务计算,E-mail:3361852202@qq.com(通信作者)。

但该模型并没有考虑加入应急任务的情况下卫星的重调度机制。文献[4]在分析卫星任务动态调度的基础上,提出了一种基于最大自由度规则的启发式算法。文献[5]根据任务安排的难易程度和相应的时间窗口数目,提出了一种基于自由度比例的启发式算法。文献[6-9]针对卫星动态调度问题,设计了一种基于任务最大灵活性和最小竞争性的启发式规则,通过反复迭代修复,用新任务替换原有任务。但上述几种方法在任务调度时没有考虑如何有效降低任务之间的相互影响,提高任务调度效益。

综上所述,现有研究在进行卫星动态重调度时,忽略了卫星资源失效的随机性和任务到达的时效性的特点,而且缺少合理的启发式规则来对任务进行融合处理,因此如何建立合理的动态重调度模型并研究高效的算法进行求解是当前亟需解决的问题。本文在考虑卫星资源失效和应急任务加入等动态不确定性因素的前提下,设计并实现了基于触发规则的事件驱动策略,构建了以最大化调度收益和最小化扰动测度为目标函数的反应式调度多目标优化模型,提出的 MISR-HA 算法有效地解决了该问题。

2 模型建立

2.1 参数定义

本文使用的参数如下: $T = \{t_1, t_2, \dots, t_{L_c}\}$ 为任务集; L_c 为任务个数; $S = \{s_1, s_2, \dots, s_{L_s}\}$ 为卫星资源集; L_s 为卫星个数; $O^n = \{o_1^n, o_2^n, \dots, o_{L_o}^n\}$ 为卫星 s_n 的轨道圈次集; L_o 为卫星 s_n 的轨道圈次数量; eot_{li}^n 为任务 t_i 在卫星 s_n 的第 L 个轨道圈次的开始时间; lot_{li}^n 为任务 t_i 在卫星 s_n 的第 L 个轨道圈次的结束时间; tac_{ij} 为任务 t_i 到任务 t_j 的姿态调整时间; p_i 为任务 t_i 的优先级; $freetime_i$ 为任务 t_i 的空闲时间; d_i 为任务 t_i 的观测时长; e_t 为卫星单位时间内所消耗的能量; e_c 为卫星姿态调整单位时间内所消耗的能量; e_s 为卫星单次姿态调整固有的能量消耗; E_n 为卫星 s_n 单个轨道的最大可用能量; c_t 为卫星单位时长所占用的存储容量; C_n 为卫星 s_n 单个轨道圈次的最大存储容量。

2.2 决策变量定义

首先考虑任务 t_i 是否能够被卫星在相应的轨道圈次进行观测,描述形式如下:

$$x_{li}^n = \begin{cases} 1, & \text{卫星 } s_n \text{ 在第 } L \text{ 个轨道圈次内对任务 } \\ & t_i \text{ 进行观测} \\ 0, & \text{其他} \end{cases}$$

其次,在不确定环境中,经常有各种干扰因素对原有调度方案造成影响。新旧方案之间的差异有 4 种,即原方案任务观测时间发生变化、任务观测窗口改变、新任务将原有任务替换和新任务无法满足观测需求。其描述如下:

$$\phi_{kr}^i = \begin{cases} 1, & \text{任务 } t_i \text{ 在第 } k \text{ 次调度过程中发生} \\ & \text{第 } r \text{ 类任务变化} \\ 0, & \text{其他} \end{cases}$$

2.3 目标函数

在卫星任务调度问题中,首先保证调度收益的最大化,其目标函数表达式如式(1)所示:

$$M_1 = \max \sum_{i=1}^{L_c} \sum_{n=1}^{L_s} \sum_{l=1}^{L_o} p_i \cdot x_{li}^n \quad (1)$$

同时,在发生外界干扰后,应尽可能减少新旧方案之间的差异,将扰动因素降到最低,将上述提到的 4 类差异变化表示如下:

设 μ_r^i 为任务 t_i 发生第 r 类任务变化后的影响因子,且 $\mu_1^i < \mu_2^i < \mu_3^i < \mu_4^i$ 。则最终最小化扰动测度如式(2)所示:

$$M_2 = \min \sum_{i=1}^{L_c} \sum_{k=1}^{L_g} \sum_{r=1}^4 \mu_r^i \phi_{kr}^i \quad (2)$$

2.4 约束条件

约束条件如下:

$$\forall t_i \in T, \sum_{l=1}^{L_o} \sum_{n=1}^{L_s} x_{li}^n \leq 1 \quad (3)$$

$$\forall t_i, t_j \in T, i \neq j \quad (4)$$

$$\sum_{l=1}^{L_o} \sum_{n=1}^{L_s} \sum_{i=1}^{L_c} x_{li}^n = \sum_{l=1}^{L_o} \sum_{n=1}^{L_s} \sum_{i=1, j=2}^{L_c} y_{ij}^{nl} \\ \forall i \in [1, L_c], \forall l \in [1, L_o], \forall n \in [1, L_s], \quad (5)$$

$$lot_{li}^n - eot_{li}^n = d_i \\ \forall t_i, t_j \in T, i \neq j, \quad (6)$$

$$y_{ij}^{nl} = 1 \rightarrow eot_{ij}^n \geq lot_{li}^n + tac_{ij} + freetime_i \\ \forall s_n \in S, \forall l \in O^n \quad (7)$$

$$\sum_{i=1}^{L_c} c_t \cdot d_i \cdot x_{li}^n \leq C_n \\ \forall s_n \in S, \forall l \in O^n \quad (8)$$

$$\sum_{t_i \in T} e_t \cdot d_i \cdot x_{li}^n + \sum_{t_i, t_j \in T, i \neq j} (e_c \cdot tac_{ij} + e_s) y_{ij}^{nl} \leq E_n$$

式(3)和式(4)表示任务 t_i 执行的唯一性约束;式(5)表示任务 t_i 一旦开始执行便不可中断;式(6)表示同一轨道相邻任务之间必须符合相应的时间间隔约束;式(7)为卫星在同一轨道存储能力的约束;式(8)为卫星同一轨道的能量约束。

2.5 基于遗传算法的初始调度方案

(1) 编码设计

本文采用“卫星资源-轨道圈次”形式定义单个染色体,其示意图如图 1 所示。

卫星-轨道	3-1	2-2	0-0	4-1	1-2
任务	1	2	3	4	5

图 1 任务调度编码示意图

Fig. 1 Schematic diagram of task scheduling coding

图 1 表示第 1 号任务分配给第 3 颗卫星的第 1 个轨道,第 2 号任务分配给第 2 颗卫星的第 2 个轨道,第 4 号任务分配给第 4 颗卫星的第 1 个轨道,第 5 号任务分配给第 1 颗卫星的第 2 个轨道。其中,“0-0”表示第 3 号任务没有对应的卫星资源进行分配。

(2) 遗传因子设计

$$f_i = \frac{p_i}{1 + \delta_i} \quad (9)$$

其中, p_i 表示任务 t_i 的优先级, δ_i 为惩罚值。惩罚值主要与约束条件有关,若任务 t_i 未满足任一约束条件,则惩罚值为 1,当出现“0-0”时惩罚值为 1,最后将惩罚值累加得到 δ_i 。

(3) 适应度函数设计

1) 选择算子

选择算子采用轮盘赌方式将优良个体保存到下一代,其

被选取的概率 P_i 为:

$$P_i = \frac{f_i}{\sum_{k=1}^{L_c} f_k} \quad (10)$$

2)交叉算子

本文选用单点交叉方式,随机选择一个交叉点交换染色体位置。

3)变异算子

变异概率 P 的计算公式为:

$$P = \frac{B}{L_c q} \quad (11)$$

其中, B 为每一代染色体中需要变异的个数, L_c 为基因个数。

(4)终止条件与解码

终止条件为遗传代数,解码与编码过程相反,最终输出初始调度方案,即将任务 t_i 分配给某颗卫星的某个轨道,“0-0”表示该任务无适合的卫星资源调度。

3 反应式动态调度驱动策略

关于反应式动态调度,需要解决的最大问题即“什么时候调度”和“采用什么驱动策略调度”。目前,常用的3个反应式调度策略包括:周期驱动、事件驱动和混合驱动。周期驱动是指在一个调度周期 T 内,无论发生任何扰动事件,都不会改变其调度方案,在下一个周期开始时再重新调度。其优点是方案简单、易于操作,但对于有时效要求的应急任务,其调度执行的滞后性将严重影响调度结果。事件驱动是指每次发生干扰事件后,原调度方案将根据需求立即进行调整,其优点是可以随时满足动态时效需求,但频繁的方案调整会带来大量额外的开销,其调度效率很难得到保障。混合驱动则结合周期驱动和事件驱动,在一定时间周期内对方案进行重新调度。

因为事件驱动能够很好地满足卫星任务调度的动态性,目前有大量文献^[10-15]采用事件驱动来确定调度时机,并通过改进时间间隔等方式提高调度效率,但仍然无法很好地解决事件驱动频繁调度对系统稳定性造成的影响。因此有必要设计一种触发规则,有效平衡调度收益最大化和扰动调度最小化。本文针对卫星任务动态调度的时效性和卫星资源失效等应急情况,设计了一种基于触发规则的事件驱动策略。其基本思想是:设计触发函数并设定触发阈值,对干扰事件进行分析,一旦达到触发阈值,就对方案进行反应式调度。设计的触发函数 TF 如式(12)所示:

$$TF = \sum_{i=1}^{t_{new}} (p_i + \frac{SC}{et_i - t}) + \beta \quad (12)$$

其中, t_{new} 表示新到达的任务数量; SC 为任务的调度周期; et_i 为用户期望任务 t_i 完成的时间; β 表示迫切参数,其触发阈值为 T_{thre} 。

由式(12)可以看出,触发函数的设计主要与以下几个因素有关:

(1)新到达任务的优先级。优先级越高,越容易达到触发阈值。

(2)新任务的时效性。由 $\frac{SC}{et_i - t}$ 可知,新任务越接近任务完成时间,即任务越紧迫,越容易触发反应式规则。

(3)紧急事件。当出现如卫星资源失效、云层遮蔽等紧急事件时,令 $\beta = T_{thre}$,即达到触发阈值从而进行重新调度,同时将失效卫星上对应的原有任务作为新任务进行下一阶段的重新调度。

对于已经达到触发阈值的反应调度,将触发函数值初始化为零;如果没有达到,则将触发函数值进行累加。

4 MISR-HA 启发式算法的求解

卫星动态调度需要面对各种干扰事件,对任务调度的时效性和优化性的要求较高。相比完全重调度以保证调度结果全局最优,局部修复的方法可以在很短的时间内得到相对较优的调度结果,调度效率高,系统相对稳定。而启发式算法则是该方法的核心,目前大量文献都采用该算法来解决卫星动态调度问题。文献^[15]利用时间窗的拥挤度和任务重叠度构造启发式因子,建立基于任务优先级的启发式插入规则。文献^[10]建立了基于循环迭代的任务替代规则。这些方法在一定程度上可以解决卫星动态调度问题,但在实际中仅考虑了新任务的插入情况,并没有考虑卫星资源失效、任务时效性等实际需求。本文在考虑实际需求的同时,结合任务选择、合成、替换策略,提出了一种融合新任务合并、插入、移位和替换的启发式算法(MISR-HA)。

4.1 任务选择策略

在反应式动态调度过程中,新任务一般指用户提交的应急任务和失效卫星上对应的原有任务,当这些新任务同时到达时,需要一定的选择策略对其进行优先级排序,本文主要使用任务完成的迫切度即根据新任务的优先级以及其可用时间窗口数目来确定。迫切度的表达式如式(13)所示:

$$\rho_i = \frac{P_i}{T_{AW}^i} \quad (13)$$

其中, T_{AW}^i 表示可用时间窗口数目,因为一个任务可以由同一个卫星不同轨道圈次观测或者由不同卫星来观测,其可用时间窗口数目越多,该任务可被调用的机会越多,其被调度的迫切度就越低。 ρ_i 表示迫切度,迫切度高者优先作为调度目标。同时,对于多个时间窗口的选择,以临近时间窗口选择为目标,这样可以在一定程度上满足应急任务对时效性目标的要求。

4.2 任务合成策略

任务合成可以使卫星以较少的开机时间完成更多的任务,若新任务通过与原任务合并可以避免冲突,则既可以增加观测收益,又可以减少卫星资源消耗。根据观测原理,卫星可以同时观测同一场景中的一些相邻目标。在本文的研究中,任务合成策略是指从已有调度方案中选择一个原任务 t_r 与新到达的任务 t_i 合并同时进行观测,从而生成一个合成任务 t_i^* 。任务合成约束的描述如下:

$$\text{s. t. } \begin{cases} \text{(a): } |\theta_{ij} - \theta_{i'j}| \leq \sigma_j \\ \text{(b): } \min\{lot_{ij}^n, lot_{i'j}^n\} - \max\{eot_{ij}^n, eot_{i'j}^n\} \geq \max\{d_i, d_{i'}\} \end{cases} \quad (14)$$

其中, θ_{ij} 是卫星 s_j 观测任务 t_i 的偏转角; $\theta_{i'j}$ 是卫星 s_j 观测任务 $t_{i'}$ 的偏转角; σ_j 为卫星 s_j 的视场角; (a)为角度约束,通过调整观测角度可以同时观测任务 t_r 和 t_i ; (b)为时间窗口约

束,两个观测任务 t_i 和 t_j 之间需要有足够的时间间隔。由以上分析可知,任何两个任务满足以上条件即可进行合并,由于任务具有多个可见机会,因此可能存在多种合并机会。

对于合成任务 t_i^* , 其偏转角为 $\theta_{ij} = \frac{\theta_{ij} + \theta_{ij}}{2}$, 时间窗为

$[eot_{hi}^*, lot_{hi}^*]$, $lot_{hi}^* = \max\{eot_{hi}^*, eot_{hj}^*\}$, $eot_{hi}^* = \min\{lot_{hi}^*, lot_{hj}^*\}$ 。另外, p_i^* 和 d_i^* 分别是复合任务 t_i^* 的优先级和观测时长, $p_i^* = p_i + p_j$, $d_i^* = \max\{d_i, d_j\}$ 。

4.3 任务替换策略

当新任务时间窗口与原任务冲突时,需要一定的替换策略判断新任务是否替换原任务。其基本思想是确保被替换的任务对其他任务的影响最大,如果替换该任务,则可以观测到许多其他任务,从而提高整体观测收益。替换任务的选择取决于评估性能,而评估性能则是基于每个任务的冲突程度 δ , 即观测时间窗口与该任务冲突的任务数量。基于冲突程度的评估函数 f_i 的定义如式(15)所示:

$$f_i = \frac{p_i}{d_i(1+\delta_i)^2} \quad (15)$$

其中, c_i 表示任务 i 的收益, δ_i 表示任务 i 的冲突程度。分母 $(1+\delta_i)^2$ 的设计旨在通过扩大惩罚来避免冲突。若新任务的评估函数值大于原任务,则用新任务替换原任务。

4.4 算法描述

结合以上分析, MISR-HA 算法主要包括 3 个主要部分: 反应式调度方案参数初始化、干扰事件分析明确触发时机和启发式规则描述。MISR-HA 算法的具体步骤如下。

步骤 1 反应式调度方案参数初始化。首先利用遗传算法^[17]生成初始预调度方案; 初始化参数影响因子 $\mu_1^i, \mu_2^i, \mu_3^i, \mu_4^i$ 和触发阈值 T_{thre} 。

步骤 2 分析外界环境扰动因素, 根据新任务优先级和时效性, 计算 $p_i + \frac{SC}{et_i - t}$ 。若出现卫星资源失效或云层遮蔽等紧急事件, 则令 $\beta = T_{thre}$, 并代入式(12)。

步骤 3 判断触发函数是否达到阈值, 重新调度已经达到触发阈值的函数, 并将触发函数值初始化为零; 如果没有达到触发阈值, 则将触发函数值进行累加。

步骤 4 计算卫星与新任务之间的可用时间窗口, 若无相应窗口, 则删除该新任务。

步骤 5 判断新任务和原方案中的调度任务能否合成, 若符合条件, 则将两个任务做合成处理, 转步骤 9; 若无法合成则转步骤 6。

步骤 6 将新任务直接插入空闲时间窗口, 判断新任务与其他任务是否冲突。若不冲突则转步骤 9, 若冲突则转步骤 7。

步骤 7 将冲突任务进行移位来插入该任务, 若插入成功则转步骤 9; 若无法进行移位插入, 则转步骤 8。

步骤 8 计算任务冲突度, 根据冲突度评估函数进行判断, 若新任务的评估函数值大于原任务, 则替换掉原任务。

步骤 9 判断是否还有新任务未进行调度, 若有则转步骤 2, 否则算法结束。

MISR-HA 算法的流程如图 2 所示。

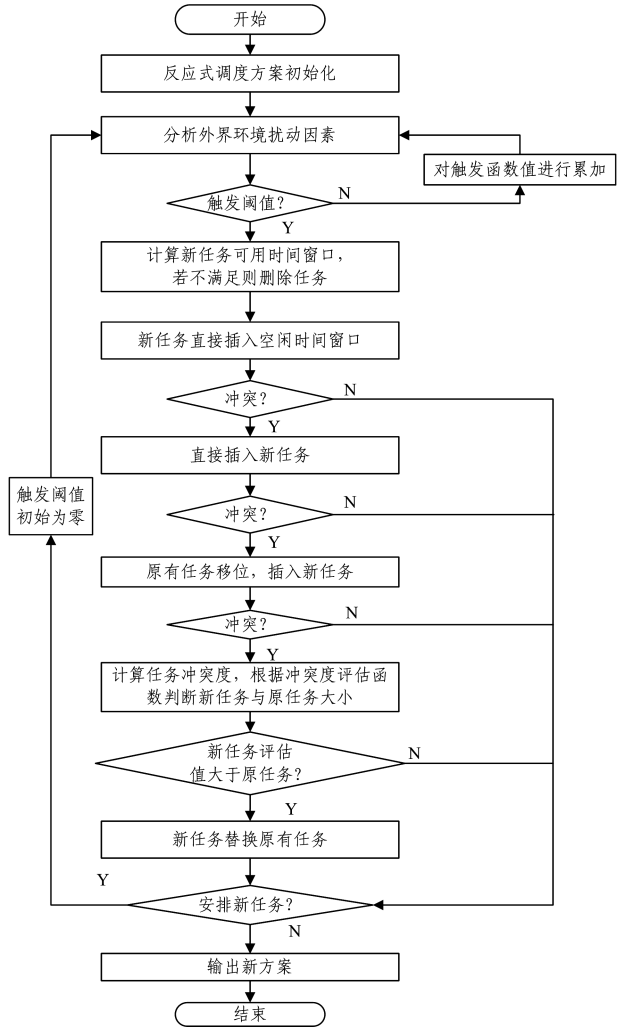


图 2 MISR-HA 算法流程

Fig. 2 Flow chart of MISR-HA algorithm

5 仿真实验

本文首先设置 6 颗卫星轨道参数, 以及初始任务、新任务等相关基本参数, 然后通过周期驱动策略、事件驱动策略进行对比分析, 来验证本文驱动策略对调度结果的影响, 最后通过 Matlab R2014b 仿真实现 MISR-HA 算法, 并与其他算法对比验证本文算法的优越性。

5.1 参数设置

针对新任务加入、卫星资源失效等动态因素, 反应式调度基于初始预调度进行重新调度。本文采用遗传算法来生成初始预调度方案, 调度周期为 24h, 初始调度任务数目为 300, 初始任务是在一定范围内随机生成的点目标任务。由于卫星资源属于稀缺资源, 其失效的概率很低, 在模拟卫星失效时根据轮盘赌法^[18]随机选择一颗卫星, 在一个调度周期内随机选择一个时机失效, 失效时长平均为 4~6h。对于新到达的应急任务, 其与原任务在同一个区域范围内, 到达时间在一个调度周期内随机选取, 完成时间为 8~12h, 其优先级为 $[0, 10]$ 内的随机数。MISR-HA 算法中初始化参数影响因子 $\mu_1^i = 0.6$, $\mu_2^i = 1.2$, $\mu_3^i = 1.8$, $\mu_4^i = 2.4$ 。本文选定 6 组随机到达的新任务, 其卫星轨道参数和反应式重调度中的基本参数设置如表

1、表2所列,其中轨道参数的设定会影响卫星的飞行轨迹,本文结合任务分布区域和卫星工具包(Satellite Tool Kit, STK)为6颗卫星设定不同的参数。

表1 卫星轨道参数

Table 1 Satellite orbit parameters

轨道参数	S1	S2	S3	S4	S5	S6
轨道周期/h	4	1.5	6	1.8	1.4	2.8
轨道高度/km	752	546	681	705	832	450
轨道倾角/(°)	97.028	55.951	68.824	97.3	98.721	78.856
近地点角/(°)	98.901	298.298	124.828	94.098	68.054	156.865
侧视角范围/(°)	±24	±33	±25	±45	±27	±30
升交点赤经/(°)	114.878	25.940	34.675	76.845	45.423	116.682

表2 反应式重调度基本参数

Table 2 Basic parameters of reactive rescheduling

编号	卫星数目	初始任务数目	新到达应急任务	失效卫星数目	失效卫星任务数目
1	6	300	40	1	20
2	6	300	80	1	24
3	6	300	160	1	18
4	6	300	240	1	16
5	6	300	320	1	22
6	6	300	480	1	19

5.2 仿真结果分析

首先,对比分析周期驱动、事件驱动策略与反应式调度中的驱动策略;然后,对比本文所提算法和其他算法在调度收益、扰动测度和运行时间上的优势。

(1)不同驱动策略对调度结果的影响

为验证本文所提基于触发规则的事件驱动策略的有效性,对比周期驱动和事件驱动,对比不同策略在6组仿真实例中的结果,以任务完成率、触发次数和响应时间为衡量目标,得到以下仿真结果。

采用事件驱动策略得到的结果如表3所列。

表3 基于事件驱动策略的结果

Table 3 Results based on event-driven strategies

编号	任务完成率	触发次数	响应时间/s
1	0.88	42	15
2	0.84	75	28
3	0.80	155	58
4	0.72	224	75
5	0.64	310	110
6	0.58	456	180

表5 在不同触发阈值下的调度结果

Table 5 Scheduling results under different trigger thresholds

触发阈值	1			2			3			4			5			6		
	任务完成率	触发次数	响应时间/s	任务完成率	触发次数	响应时间/s	任务完成率	触发次数	响应时间/s	任务完成率	触发次数	响应时间/s	任务完成率	触发次数	响应时间/s	任务完成率	触发次数	响应时间/s
30	0.88	12	13	0.81	18	20	0.80	48	35	0.70	84	70	0.65	132	92	0.61	135	120
60	0.84	12	13	0.80	15	18	0.79	46	32	0.68	82	68	0.64	129	90	0.60	132	118
90	0.82	11	12	0.78	15	18	0.77	42	30	0.66	72	60	0.62	120	84	0.59	130	116
120	0.79	11	12	0.74	14	18	0.73	42	30	0.64	71	60	0.60	109	79	0.58	120	110
150	0.78	11	12	0.73	14	18	0.72	40	28	0.62	60	51	0.60	90	66	0.57	99	98
180	0.76	11	12	0.72	13	26	0.71	40	28	0.60	48	40	0.58	84	59	0.56	90	90
210	0.76	10	12	0.72	13	16	0.71	32	22	0.58	36	32	0.56	60	45	0.52	66	75
240	0.70	10	12	0.69	12	16	0.68	24	18	0.56	27	25	0.54	48	39	0.52	60	72
270	0.70	9	12	0.68	10	14	0.64	16	16	0.54	18	20	0.52	36	32	0.50	54	69
300	0.66	8	12	0.65	9	14	0.59	14	15	0.50	15	18	0.49	30	30	0.48	45	65

由表5可以看出:

1)当触发阈值逐渐增大时,任务完成率和触发次数都在

由表3可以看出:

1)在事件驱动条件下,任务完成率均大于0.5,这是因为事件驱动策略下,有扰动事件发生后,即刻启动反应式调度,能够很大程度上满足任务需求。

2)随着新到达的应急任务的增加,触发次数也不断增加,这是因为当一个或者多个任务到达时,依据事件触发规则均要做出响应,因此触发次数和新到达的应急任务数目非常接近。

3)随着新到达的应急任务的增加,响应时间也在急剧增加,这是因为频繁的规则响应需要不断地重新调度,从而消耗大量的时间。

基于周期驱动策略的反应式调度需要确定触发周期,因为本文中的调度周期为24h,参考文献[19],设定触发周期为0.6h,在6组仿真实例下其调度结果如表4所列。

表4 基于周期驱动策略的调度结果

Table 4 Scheduling results based on periodic drive strategy

编号	任务完成率	触发次数	响应时间/s
1	0.75	40	10
2	0.69	40	12
3	0.64	40	16
4	0.56	40	18
5	0.47	40	22
6	0.40	40	24

由表4可以看出:

1)在周期驱动下,任务完成率随新到达任务的增加而降低,这是因为触发周期固定,当新任务较少时,可以在较短时间内安排任务;但随着任务数目的增加,因为未达到触发时机,新到达任务不断累加,有些任务会因为没有得到及时调度而错过调度时机。

2)由于触发周期固定,在一个调度周期24h内,触发次数为 $24/0.6=40$ 。

3)随着新到达任务数目的增加,响应时间并没有太大变化,这是因为基于周期驱动策略的响应时间和触发次数有很大的相关性。

对于基于触发规则的事件驱动策略,触发阈值的确定对调度结果的影响很大,本文通过仿真实例来确定最为合适的触发阈值。设定触发区间为 $[30, 300]$,间隔为30,在6组仿真实例下其调度结果如表5所列。

逐渐减小。这是因为当触发阈值较小时,新任务很容易达到临界值,任务调度比较频繁,任务完成率较高;但随着触发阈

值逐渐增大,相邻两次调度之间的间隔时间也会变长,触发次数逐渐减少,越来越多的任务无法在规定时间内窗口内完成调度,从而导致任务调度率降低。

2)由表中数据对比分析可知,当目标的任务完成率较高时往往触发阈值较低,但频繁的触发反应式调度会导致系统不稳定且资源消耗较大;当目标的触发次数较少时往往触发阈值较高,但任务完成率一般相对较低。5表中触发阈值在[180,240]之间时,任务完成率普遍在0.5以上,且触发次数

较低,调度结果可以接受,因此本文选择210作为触发阈值。

3)基于触发规则的事件驱动策略的反应时间在2min之内,随着触发阈值的增加,触发次数减少,反应时间也逐渐缩短。

由以上分析可得:在不同驱动策略的影响下,任务完成率、触发次数和响应时间这3个衡量指标都会有所不同。为了进一步证明本文所提策略的有效性,将以上3种驱动策略做纵向对比分析,其中周期驱动策略触发周期为0.6h,事件驱动策略触发阈值为210,其对比结果如表6所列。

表 6 不同驱动策略下 3 个衡量指标的对比结果

Table 6 Comparison results of three metrics under different driving strategies

编号	事件驱动			周期驱动			基于触发规则的事件驱动		
	任务完成率	触发次数	响应时间/s	任务完成率	触发次数	响应时间/s	任务完成率	触发次数	响应时间/s
1	0.88	42	15	0.75	40	10	0.76	10	12
2	0.84	75	28	0.69	40	12	0.72	13	16
3	0.80	155	58	0.64	40	16	0.71	32	22
4	0.72	224	75	0.56	40	18	0.58	36	32
5	0.64	310	110	0.47	40	22	0.56	60	45
6	0.58	456	180	0.40	40	24	0.52	66	75

由表 6 可以看出:

1)本文所提驱动策略的任务完成率相比事件驱动策略平均降低了14.01%左右,相比周期驱动平均提高了9.64%左右。

2)本文所提驱动策略的触发次数相比事件驱动策略平均降低了80.83%左右,相比周期驱动平均降低了9.58%左右。

3)本文所提驱动策略的响应时间相比事件驱动策略平均快49.83%左右,相比周期驱动平均慢38.67%左右。

综上所述,虽然本文所提驱动策略在任务完成率方面略低于事件驱动策略,但在响应时间方面却具有明显优势,且在触发次数方面全面优于其他两种策略。因此,实验结果证明了该策略的有效性。

(2)算法对比结果分析

为了进一步验证本文 MISR-HA 算法的优越性,将其与文献[4]中基于最大自由度的启发式算法(Max-Flexibility based Heuristic Algorithm, MF-HA),以及文献[5]中基于自由度比例的启发式算法(Flexibility Proportion based Heuristic Algorithm, FP-HA)和只考虑任务插入的启发式算法(Task Inserting based Heuristic Algorithm, TI-HA)进行对比。算法的实验环境为 Lenovo G510 和 2.6 GHz 双核 CPU, 12.0 GB 内存,操作系统为 Windows 10 X64,采用 Matlab R2014b 实现编程。为消除少数实验的随机性造成的影响,本文将每种算法迭代 20 次,以调度收益、扰动测度和平均运行时间为最后对比的参考结果,具体结果如图 3—图 5 所示。

由图 3 可知,随着新到达任务数目的增加,4 种算法的调度收益都在不断增加。其中, MISR-HA 算法的调度收益在 4 种算法中最高,比 MF-HA, FP-HA, TI-HA 算法的平均收益分别增加了 10.43%, 13.15% 和 20.77%。由于 TI-HA 算法仅考虑任务插入,许多任务因无法直接插入而被舍去,因此该算法的收益最少。同时,相比于 MF-HA 和 FP-HA 算法,本文算法考虑了合成因素,减少了相互排斥的任务数目,从而有效地提高了调度收益。

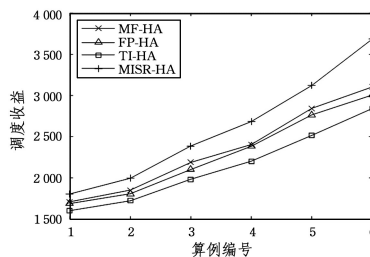


图 3 MISR-HA 与 MF-HA, FP-HA, TI-HA 算法的调度收益对比
Fig. 3 Comparison of scheduling revenues among MISR-HA, MF-HA, FP-HA and TI-HA algorithm

由图 4 可知,随着新到达任务数目的增加,4 种算法的扰动测度也在不断增加。这是因为随着调度规模的增加,越来越多的任务需要根据时间窗口进行不断的调整,不断有任务被移位和替换。其中, MISR-HA 算法的扰动测度相比 MF-HA, FP-HA, TI-HA 算法的平均扰动测度分别降低了 40.78%, 28.76% 和 56.19%。MISR-HA 算法有效保证了系统调度的稳定性。

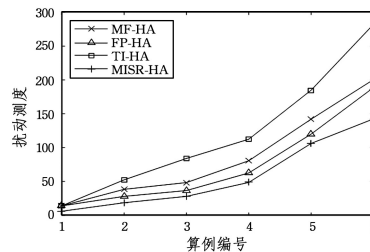


图 4 MISR-HA 与 MF-HA, FP-HA, TI-HA 算法的扰动测度对比
Fig. 4 Comparison of disturbance measurement among MISR-HA, MF-HA, FP-HA and TI-HA algorithm

由图 5 可知,随着新到达任务数目的增加,4 种算法的平均运行时间也在不断增加。其中, MISR-HA 算法的平均运行时间相比 MF-HA 和 FP-HA 算法分别缩短了 37.70% 和 33.88%, 相比 TI-HA 算法延长了 27.69%。这是因为 TI-HA 算法没有考虑任务合并、移位和替换等策略规则,其调度

相对简单,运行时间最短。MISR-HA 算法在 6 组算例中均能在 75 s 内得到调度结果,调度效率较高。

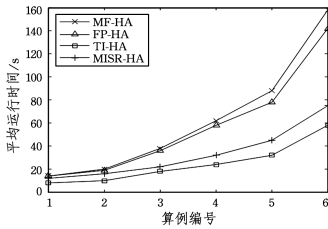


图 5 MISR-HA 与 MF-HA,FP-HA,TI-HA 算法平均运行时间对比示意图

Fig. 5 Comparison of average running time of MISR-HA, MF-HA, FP-HA and TI-HA algorithms

综上所述,MISR-HA 算法能够在比较短的时间内得到调度收益高、扰动测度小的调度结果,证明了该算法的可行性。针对新任务到达、卫星资源失效等不确定性因素,本文研究的基于启发式算法的卫星反应式调度能够有效解决此类动态问题。

结束语 针对卫星资源失效和应急任务加入两类不确定因素,在考虑多种约束的前提下,本文构建了以最大化调度收益和最小化扰动测度为目标函数的反应式调度多目标优化模型,通过计算触发函数值是否达到触发阈值来判断是否启动反应式调度,进而提出了一种基于触发规则的事件驱动策略,以及一种考虑任务合成、插入、移位、替换的启发式算法,最后通过仿真算例分析了驱动策略的影响,验证了所提算法的优越性。

参 考 文 献

- [1] WANG J,ZHU X,YANG L T,et al. Towards dynamic real-time scheduling for multiple earth observation satellites[J]. Journal of Computer and System Sciences, 2015, 81(1): 110-124.
- [2] GRASSET-BOURDEL R,VERFAILLIE G,FLIPO A. Planning and replanning for a constellation of agile Earth observing satellites[J]. Proceedings of International Conference on Automated Planning and Scheduling, 2011, 26(2): 68-86.
- [3] HAO H C,JIANG W,LI Y J. Dynamic Task Planning Problem Based on Multi-Agent Agile Satellite[J]. Journal of National University of Defense Technology, 2013, 35(1): 53-59. (in Chinese)
郝会成,姜维,李一军. 基于 Multi-Agent 敏捷卫星动态任务规划问题[J]. 国防科技大学学报, 2013, 35(1): 53-59.
- [4] WANG J M,LI J F,TAN Y J. Study on multi-star dynamic scheduling model and algorithm with new task insertion[J]. Journal of System Simulation, 2009, 21(12): 3522-3527. (in Chinese)
王军民,李菊芳,谭跃进. 有新任务插入的多星动态调度模型与算法研究[J]. 系统仿真学报, 2009, 21(12): 3522-3527.
- [5] PEMBERTON J C, GREENWALD L G. On the need for dynamic scheduling of imaging satellites [J]. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 2002, 34(1): 165-171.
- [6] KRAMER L A, SMITH S F. Maximizing Flexibility: A Retraction Heuristic for Oversubscribed Scheduling Problems [C] // Proceedings of the 18th International Joint Conference on Artificial Intelligence. Acapulco Mexico, 2003.
- [7] KRAMER L A, SMITH S F. Task Swapping for Schedule Improvement: A Broader Analysis [C] // Proceedings of the 14th International Conference on Automated Planning and Scheduling. Whistler, Canada, 2004.
- [8] KRAMER L A, SMITH S F. Task Swapping: Making Space in Schedules for Space [C] // Proceedings of the 4th International Workshop on Planning and Scheduling for Space. Darmstadt, Germany, 2004.
- [9] KRAMER L A, SMITH S F. Maximizing flexibility: A retraction heuristic for oversubscribed scheduling problems [C] // IJ-CAI. 2003: 1218-1223.
- [10] FOX M, GEREVINI A, LONG D, et al. Plan Stability: Replanning versus Plan Repair [C] // Proceedings of the International Conference on Automated Planning and Scheduling, 2006: 212-221.
- [11] LI Y Q, WANG R X, XU M Q. Scheduling and Rescheduling of Imaging Satellite Based on Ant Colony Optimization [J]. Journal of Computational Information Systems, 2013, 9(16): 6503-6510.
- [12] LIU Y. Research on Dynamic Rescheduling Model, Algorithm and Application of Reconnaissance Satellite [D]. Changsha: National University of Defense Technology, 2005. (in Chinese)
刘洋. 侦察卫星动态重调度模型、算法及应用研究 [D]. 长沙: 国防科技大学, 2005.
- [13] ZHU J H, LIU J. Local Dynamic Collaborative Rescheduling Method for Agile Satellite Observation Scheme [J]. Military Planning and Systems Engineering, 2014, 28(3): 48-52. (in Chinese)
祝江汉, 刘进. 敏捷卫星观测方案局部动态协同重调度方法 [J]. 军事运筹与系统工程, 2014, 28(3): 48-52.
- [14] WANG J, ZHU X, QIU D, et al. Dynamic scheduling for emergency tasks on distributed imaging satellites with task merging [J]. IEEE Transactions on Parallel and Distributed Systems, 2014, 25(9): 2275-2285.
- [15] WANG J, ZHU X, YANG L T, et al. Towards dynamic real-time scheduling for multiple earth observation satellites [J]. Journal of Computer and System Sciences, 2015, 81(1): 110-124.
- [16] BILLUPS S C. Satellite Mission Scheduling with Dynamic Tasking [R]. Final Report of the UCDHSC Mathematics Clinic, 2005.
- [17] YANG H T, YANG L, ZHAO H L. Study on the mission planning model of space-based information acquisition [J]. Journal of System Simulation, 2009, 21(21): 6710-6715.
- [18] FRANCOIS V, ARDA Y, CRAMA Y, et al. Large neighborhood search for multi-trip vehicle routing [J]. European Journal of Operational Research, 2016, 255(2): 422-441.
- [19] CHURCH L K, UZSOY R. Analysis of periodic and event-driven rescheduling policies in dynamic shops [J]. International Journal of Computer Integrated Manufacturing, 1992, 5(3): 153-163.