

资源约束突变的航天器观测快速重调度优化算法

赵凡宇^{1,2}, 徐瑞^{1,2}, 崔平远^{1,2}

(1. 北京理工大学 深空探测技术研究所, 北京 100081; 2. 飞行器动力学与控制教育部重点实验室, 北京 100081)

摘 要:针对航天器对地观测调度中资源约束发生突变的情况,提出了一种基于蚁群算法的启发式重调度算法。首先对重调度过程中的资源约束进行分析,给出了资源约束发生变化的重调度模型。然后,结合原调度优化结果,给出重调度任务集合更新方法,对任务集合进行剪裁。最终,基于最大限度利用原调度方案信息的思想,结合任务集合更新及优先级等启发式信息,给出了一种改进的重调度优化算法。数值计算结果表明,所设计的算法可以快速有效的提高重调度过程的收益。

关键词: 航天器观测; 重调度; 蚁群优化; 资源约束

中图分类号: V19

文献标识码: A

文章编号: 2095-7777(2015)03-0262-05

DOI:10.15982/j.issn.2095-7777.2015.03.012

0 引 言

在地面观测需求日益增多的今天,为了提高对地观测航天器的工作效率和运行成本,需要对其观测任务进行合理的调度。空间环境的恶劣及不确定性,使得航天器自身并不能一直保持稳定的状态。当航天器的状态发生突变时,其既定的调度方案可能不再适应原有的观测需求,此时则需要对航天器的观测任务进行重新调度。原有的调度模型只是进行了部分改变,此时若不充分利用已有的调度方案信息,则将有可能增大重新调度过程的计算量。因此,重新调度的过程需要最大限度的利用已有调度方案信息,从而提高重调度问题的求解速度和目标收益。本文针对利用原有调度方案信息重调度问题进行研究。

关于航天器观测重调度问题的研究,目前进行得比较少。德国 Varfaillie 和 Schiex(1994)在研究欧洲航天局 SPOT 卫星调度问题(单星调度)时,针对新任务到达的情况,提出了一种动态的处理思想^[1],即一个新的观测任务能够插入到调度方案中的充分必要条件是:如果由于该任务插入而改变了初始调度方案中某些任务的初始调度结果,改变的任务必须能够在调度截止时间内重新安排位置,而不会引起冲突。这种思想事实上是以保证卫星应用的服务质量为主要目标,即一旦任务列入方案,就一定

定要完成。

Pemberton 等(2002)对多观测卫星动态调度需求进行了分析^[2],将多观测卫星动态调度需求分为四种,包括卫星资源状态的变化、新任务的到达、任务机会的选择和环境不确定性的影响等,论述了该问题作为一种连续规划/调度问题的一般特点,指出连续两个方案之间的差距需最小的求解要求。

NASA 的 Lina Khatib 等(2003)在研究航天器自主调度问题时,也提出了处理不确定事件的需求^[3],即在设计观测任务调度系统时,需要考虑观测的气象条件、敏感目标的捕获以及航天器或地面站健康状态等动态不确定因素。

刘洋(2004)在其博士论文中研究了卫星故障和新任务到达情况下的航天器动态调度问题,提出了相应的启发式算法^[4]。王军民等(2007)在对多星联合动态调度问题的动态来源进行深入分析的基础上,对该问题进行了统一描述,并提出了一种基于规则的启发式求解算法^[5]。李玉庆等(2012)研究了基于时间约束的受影响任务分析算法,并给出了基于局部信息素调整的重调度算法^[6]。

综上所述可知,大部分重调度研究集中在突发事件影响时序约束的调度问题。对于时间约束受影响任务,通过分析进行剔除或增加不确定性分析,可以有效地利用原调度信息。然而对资源约束发生突变时,重调度过程一般进行生成式重调度,而不能有

效利用已有信息,达到快速重调度优化的目的。本文针对资源约束发生变化时,航天器观测重调度问题进行研究,基于最大限度利用原有调度方案的原则,对任务集合进行剪裁,提出一种修正式重调度方法。

1 重调度模型

航天器对地观测的调度过程具有多目标任务多资源约束,一般可称之为多任务多资源调度问题(multi-task and multi-resource scheduling, MTRS)。本文研究对地观测目标任务具备不同优先级、资源约束具有多类型特性的调度问题。

为了方便研究计算,并考虑实际情况,本文作如下假设:

1)低轨卫星进行对地观测时,观测目标都为条状目标,且平行于星下线;

2)单个轨道圈次内电量总量为常数,在每个轨道圈次开始时,经过调度时段之外的轨道弧段上的补充,调整为固定值;单一对地观测时间所需时间为常数;

3)重调度过程发生的前提条件为:原调度优化过程结束之后,检测到资源约束发生突变。

1.1 模型中的参数定义

多任务多资源问题可以描述为六元组的形式

$$MTRS = \{S, T, P, G, R, W\} \quad (1)$$

其中: S 表示卫星集合,设可以执行观测任务的卫星数量为 N_S ,有 $S = \{s_1, s_2, \dots, s_{N_S}\}$; T 表示所有任务集合,设候选观测任务数量为 N_T , $T = \{t_1, t_2, \dots, t_{N_T}\}$,对候选观测任务 t_i , w_i 为任务 t_i 的优先级; E 表示调度过程中能量约束的集合; W 表示可见窗口集合。

时序参数: obt_{mi}^p 为选中的当前任务的起始时间; d_i 为候选任务 t_i 的观测持续时长; tac_{ij} 为任务 t_i 到任务 t_j 的姿态调整时间长度。

优先级参数: w_i 为任务 t_i 的优先级。

电量参数: E_s^p 为卫星 s_p 单个轨道圈次内可用的电量总量; e_i 为任务 t_i 观测所需消耗电量; e_i^d 为数据下传任务 td_i 所需消耗电量; e_M 为姿态机动单位角度所需能量; e_s 为姿态稳定所需消耗电量,常数。

调度过程中的目标选择标志变量为

$$c_{mi}^p = \begin{cases} 1 & tp_i \text{ 由卫星 } s_p \text{ 在轨道圈次 } ot_m^p \text{ 内执行} \\ 0 & tp_i \text{ 未由卫星 } s_p \text{ 在轨道圈次 } ot_m^p \text{ 内执行} \end{cases} \quad (2)$$

$$c_{ij}^{Pm} = \begin{cases} 1 & t_j \text{ 是卫星 } s_p \text{ 在轨道圈次 } ot_m^p \text{ 内} \\ & \text{执行 } t_i \text{ 的后续任务} \\ 0 & t_j \text{ 不是卫星 } s_p \text{ 在轨道圈次 } ot_m^p \text{ 内} \\ & \text{执行 } t_i \text{ 的后续任务} \end{cases} \quad (3)$$

1.2 重调度模型

综合考虑各类约束,建立约束模型。

调度的目标函数为

$$P = \sum_{i=1}^{N_T} \sum_{m=1}^{N_O} \sum_{p=1}^{N_S} w_i c_{mi}^p c_i \quad (4)$$

约束条件

$$\sum_{i=1}^{N_T} \sum_{m=1}^{N_O} \sum_{p=1}^{N_S} ct_{ij}^{pm} = \sum_{r=1}^{N_T} \sum_{m=1}^{N_O} \sum_{p=1}^{N_S} ct_{jr}^{pm} = \sum_{m=1}^{N_O} \sum_{p=1}^{N_S} c_{mj}^p \quad (5)$$

$$c_{ij}^{Pm} = 1, obt_{mj}^p \geq obt_{mi}^p + d_i + tac_{ij} \quad (6)$$

$$\sum_{t_i \in (T)} (e_m tac_{ij} + e_s) ct_{ij}^{pm} + e_i c_{mi}^p \leq E_s^p \quad (7)$$

其中:式(5)为观测任务执行唯一性约束;式(6)为同轨道圈次任务执行时间约束;式(7)卫星单一轨道圈次内电量约束。

2 重调度算法

2.1 任务集合更新

调度过程中的资源约束并不能满足把所有任务都安排到调度方案中。故原有的调度优化方案,是在假设满足所有约束的条件下,以获取尽可能大的收益为目的得到的。所以在资源约束发生变化之后,根据原有的调度方案信息,进行受影响任务分析,并进一步进行任务集合的裁剪,亦即搜索空间的裁剪,成为一种进行修正式重调度的手段。当资源约束发生变化时,利用原有调度方案对待选目标任务集合进行修改,使用原调度优化方案作为重调度过程的待选任务集合。

2.2 重调度优化算法

针对调度问题特性,提出一种改进的蚁群算法,用一只蚂蚁表示一个轨道圈次内的卫星资源。借鉴蚁群系统(ACS)和最大最小蚂蚁系统(MMAS)的思想设计状态转移规则和信息素更新策略,并对基本算法进行改进,结合观测约束,引入能量需求预估和容量需求预估来控制转移概率。

2.2.1 状态转移规则

$$j = \begin{cases} \max_{j \in allowed_k(t_i)} \{(\tau_{ij})^\alpha (\eta_{ij})^\beta (\omega_j)^\gamma (\vartheta_j)^\epsilon\} & q \leq q_0 \\ S & q > q_0 \end{cases} \quad (8)$$

式中, $\text{allowed}_k(t_i) = T - \sum_{i=1}^{N_O N_S} \text{tabu}_i - t_violate_k$ 为蚂

蚁 k 下一步允许执行任务的集合。其中: $\sum_{i=1}^{N_O N_S} \text{tabu}_i$ 为所有蚂蚁已经探索过的任务目标, 此集合为全局变量。在飞行过程中, 卫星的位置是以时间为变量的函数值, 所以需要考虑目标之间的时序约束; $t_violate_k$ 为当前蚂蚁 k 位置时间参数不能满足时序约束的目标任务集合。

q_0 为控制转移规则的参数, 在 $[0, 1]$ 区间内取值; q 为 $[0, 1]$ 区间上均匀分布的随机变量。S 根据下式进行选择

$$p_{ij}^k = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta (\omega_j)^\gamma (\vartheta_j)^\epsilon}{\sum_{l \in \text{allowed}_k(t_i)} \tau_{il} (\eta_{il})^\beta (\omega_l)^\gamma (\vartheta_l)^\epsilon} & j \in \text{allowed}_k(t_i) \\ 0 & \text{其它} \end{cases} \quad (9)$$

其中: τ_{ij} 为任务 t_i 与任务 t_j 间的信息素强度; $\eta_{ij} = \frac{1}{\xi_{ij}}$, $\xi_{ij} = \text{tac}_{i,j} + e_j$ 为任务 t_i 与任务 t_j 间能量需求影响; ω_j 为任务 t_j 优先级的影响。 ϑ_j 为任务 t_i 与任务 t_j 之间任务转换时间值, $\vartheta_j = \frac{1}{\psi_{ij}}$, $\psi_{ij} = \text{obt}_{mj}^p - (\text{obt}_{mi}^p + d_i + \text{tac}_{ij})$ 。

2.2.2 信息素更新策略

文中的蚁群算法采用所有蚂蚁同步迭代的转移方式。首先为所有蚂蚁随机分配初始目标, 所有蚂蚁依次选择下一任务, 若无法找到任务, 则返回该蚂蚁的可行任务序列 t_f_i 。当所有的蚂蚁完成任务序列构造, 可得到可行任务序列集合为

$$TF = \bigcup_{i=1}^{N_O N_S} t_f_i \quad (10)$$

将其收益与当前最优任务执行序列集 TF_{Best} 作比较。

信息素的更新。若 TF 的总收益, 即全部蚂蚁探索任务的目标函数值 (任务的优先级与获得数据量的乘积之和) P_{TF} 不大于 TF_{Best} 的总收益 P_{Best} , 则 TF 所有边上的信息素挥发。经过挥发更新之后, 信息素浓度为: $\tau_{ij}^{\text{new}} = (1 - \rho) \tau_{ij}^{\text{old}}$, 其中 ρ 为信息素挥发度, τ_{ij}^{old} 为任务 t_i 与任务 t_j 间的原信息素强度。若新生成任务序列的总收益 P_{TF} 大于 P_{Best} , 则更新全局最优解 $TF_{\text{Best}} = TF$ 。为加速收敛, 每次循环后, 只对当前全局最优解中任务序列上的信息素浓度进

行更新得

$$\tau_{ij}^{\text{new}} = (1 - \rho) \tau_{ij}^{\text{old}} + \Delta \tau_{ij} \quad (11)$$

式中 $\Delta \tau_{ij} = \frac{Q}{L_{\text{full}} - L_{\text{Best}}}$, 其中 Q 为预设参数, L_{Best} 为任务执行时资源参数的影响, 且

$$L_{\text{Best}} = \sum_{t_i, t_j \in TF_{\text{Best}}} \xi_{ij} \quad (12)$$

为了避免算法过早收敛于局部最优解, 将各条路径可能的信息素浓度限值在 $[\tau_{\min}, \tau_{\max}]$ 。

2.2.3 算法实现

算法的具体计算流程如图 1 所示。本文设计重调度算法共包括 4 部分: 重调度模型的初始化、观测任务序列生成过程、信息素更新过程及最优解的判断过程。

- Step1. 统计原调度方案中未选择目标并标记;
- Step2. 在原有目标集合中, 剔除 step2 中标记的目标;
- Step3. 更新约束;
- Step4. 更新待选目标集合, 并完成信息素的标记;
- Step5. 令 $nc=0$ (nc 为迭代次数), 将 m 只蚂蚁置于 N 个节点上, 将初始节点置于当前解集中;
- Step6. 对当前蚂蚁 k ($k \in \{1, 2, 3, \dots, m\}$), 计算满足时间约束的待选任务集合;
- Step7. 按照概率 $p_i^k(t)$ 选择下一节点; $k \in \{1, 2, 3, \dots, m\}$
- Step8. 并进行资源检查, 若无冲突则将任务 i 置入当前解集, 否则将节点 i 从待选任务集合中删除, 并转至 step7;
- Step9. 若未完成所有待选任务的遍历则 $k=k+1$, 转 step6;
- Step10. 计算各蚂蚁的收益 L_k , 记录当前最好解;
- Step11. 更新各节点信息素;
- Step12. 置 $nc=nc+1$;
- Step13. 若 $nc <$ 预定的迭代次数, 且无退化行为 (即找到的都是相同解), 则转 step5;
- Step14. 输出当前最好解。

图 1 重调度算法流程

Fig. 1 Rescheduling algorithm

算法在每一步的任务选择过程中, 通过时序约束的判断来形成候选任务集合, 在每个任务的选择过程中判断资源约束是否满足约束条件, 从而保证任务的有效性。本文即采用此种方式设置终止条件。设置算法最大遗传代数为 N_{\max} , 当蚁群迭代代数达到 N_{\max} 时算法停止。

3 算例及分析

算例中, 在 $(0 \sim 95^\circ \text{E}, 0 \sim 50^\circ \text{N})$ 范围内随机生成条状目标观测任务。任务的优先级为 $[1 \sim 10]$ 的随机数。算法用 Matlab 2010b 在 2.6 GHz CPU、4 GB 内存的计算机上实现。参数的取值如表 1 所示。

表 1 仿真过程中的各参数设置

Table 1 Parameters setting during simulation experiment

N_T	N_{max}	α	β	λ	φ	q_0	ρ	Q	τ_{max}	τ_{min}
100	100	1	3	2	1	0.5	0.1	150	1.0	0.01

图 2 给出了使用重调度优化之后产生的任务序列。图中条状图形即优化结果中对对应航天器选择的地面观测目标。横坐标为选中目标的经度位置及其覆盖地面经度范围。由位置关系可以直观看出,优化结果满足时间约束。

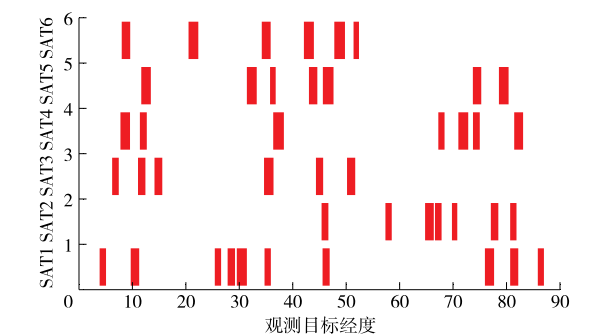


图 2 重调度之后的调度方案
Fig. 2 Rescheduling results

图 3 给出了优化过程的对比。图中上方虚线代表本文重调度优化过程,而下方实线代表全局式重调度过程。两优化过程的初值都由单次迭代过程中的任务选择过程产生,本文算法在初值求解上即已具备一定的优势,初值解观测收益比全局式重调度算法高出了 13%。而由整体优化过程,可以清晰得看出,本文算法的收敛性和优化结果都要优于全局式优化算法。

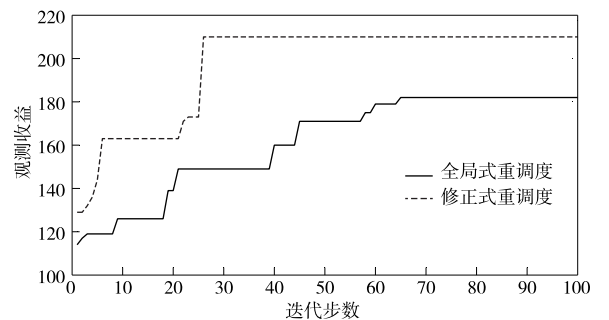


图 3 优化过程对比
Fig. 3 Optimization process comparison

4 结 论

本文给出了资源约束发生变化的重调度模型。基于最大限度利用原调度方案信息的思想,结合任务集合更新及优先级等启发式信息,给出了一种改进的重调度优化算法。数值计算结果表明,本文设计的算法可以快速有效的提高重调度过程的收益。本文研究内容有助于提高真实卫星任务调度中的数据收集能力和应用水平,为资源发生变化条件下的重调度优化过程提供参考。

参 考 文 献

[1] Varfaillie G, Schiex T. Solution reuse in dynamic constraint satisfaction problem [C] // Proceedings of the Twelfth Conference of the American Association of Artificial Intelligence. American :[s. n.],1994:307 - 312.

[2] Pemberton J C, Greenwald L G. On the need for dynamic scheduling of the image satellite [C] // Pecora15/Land Satellite information IV/ISPRS Commission I/FIEOS 2002 Conference Proceedings. [S. l.]: ISPRS, 2002.

[3] Khatib L, Frank J, Smith D, et al. Interleaved observation execution and rescheduling on Earth observing systems[C] // Proceedings of the ICAPS Workshop on Plan Execution. [S. l.]: ICAPS, 2003.

[4] 刘洋. 成像侦查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, 2004.

[5] 王军民,谭跃进. 多星联合动态调度问题的启发式算法研究[J]. 计算机工程与应用, 2007, 43(21): 21 - 25. [Wang J M, Tan Y J. Research on heuristic algorithm for problem of multi-satellites dynamic scheduling [J]. Computer Engineering and Applications, 2007, 43(21): 21 - 25.]

[6] 李玉庆,王日新,许敏强. 基于改进遗传算法的一类多资源测控调度问题研究[J]. 宇航学报, 2012, 33(1): 85 - 90. [Li Y Q, Wang R X, Xu M Q. An improved genetic algorithm for a class of multi-resource range scheduling problem [J]. Journal of Astronautics, 2012, 33(1): 85 - 90.]

作者简介:

赵凡宇(1987—),男,博士研究生,飞行器设计专业,研究方向:航天器自主任务规划与调度。
通信地址:北京理工大学宇航学院 22 号信箱(100081)
E-mail:zfybit@bit.edu.cn

Rescheduling Optimization for Spacecraft Observation with Resource Constraints Changing

ZHAO Fanyu^{1,2}, XU Rui^{1,2}, CUI Pingyuan^{1,2}

(1. Institute of Deep Space Exploration Technology, Beijing Institute of Technology, Beijing 100081, China;

2. Key Laboratory of Dynamics and Control of Flight Vehicle, Ministry of Education, Beijing 100081, China)

Abstract: A rescheduling optimizing algorithm based on ant colony optimization (ACO) is proposed in this paper for the observing rescheduling with resource constraints changing. First, the resource constraints in the rescheduling process are analyzed, and a rescheduling model with resource constraints changing is established. Second, taking the advantages of the original scheduling results, an updating method is given out for the selecting of the missions. Finally, based on the principle of taking advantage of the original scheduling results as much as possible, combining the mission updating method and priorities of the missions, a heuristic rescheduling optimizing algorithm is proposed. The results show that the algorithm could effectively improve the profit of the rescheduling process.

Key words: spacecraft observing; rescheduling; ant colony optimization; resource constraints

[责任编辑:宋宏]