航 空 学 报
Acta Aeronautica et Astronautica Sinica
May 25 2013 Vol. 34 No. 5 1151-1164
ISSN 1000-6893 CN 11-1929/V
http://hkxb.buaa.edu.cn hkxb@buaa.edu.cn
doi: 10.7527/S1000-6893.2013.0204

面向应急条件的多星动态调度方法

王建江[1], 朱晓敏[1,*], 吴朝波[1,2], 邱涤珊[1]

1. 国防科学技术大学 信息系统工程重点实验室, 湖南 长沙 410073
2. 中国人民解放军 95246 部队, 广西 北海 536000

摘 要: 针对应急条件下多星动态调度问题,建立了多目标数学规划模型,提出了应急条件多星成像任务合成策略:建立多星多轨任务合成图(MSMOTMG)模型,提出任务合成算法 CP-TM。为克服合成导致任务成像机会减少的缺陷,提出了基于合成任务分解的修复技术。此外,为进一步提高调度效率,考虑了任务在等待队列中的向后移位策略,提出了综合考虑任务合成、修复和向后移位的多星动态应急调度(TMRBS-DES)算法。通过大量模拟实验,将 TMRBS-DES 算法同 RBHA 算法,以及 3 个 baseline 算法(BS-DES、TMR-DES 和 TMBS-DES)进行了比较。实验结果表明 TMRBS-DES 算法提高了调度质量,适用于应急条件下多星动态调度问题。

关键词: 成像卫星; 动态应急调度; 数学模型; 任务合成; 向后移位; 修复; 启发式算法

中图分类号: V474; TP751.1 **文献标识码**: A **文章编号**: 1000-6893(2013)05-1151-14

成像卫星是搭载了光学传感器按照用户需求对地表特定区域进行成像的卫星平台[1-2]。成像卫星具有覆盖范围广、连续监视时间长和不受空域国界限制等优势,被广泛应用于地球资源勘测、战场态势侦察和自然灾害监视等领域[3]。同时,卫星成像已经成为应急条件下获取地面信息的重要手段[4]。应急环境(例如地震、洪涝、火灾、恐怖袭击和局部战争等)下,事件发生具有突然性,时间、地点和规模具有不确定性。此时,为及时开展救援行动,成像卫星需要在几小时、甚至几十分钟内提供服务[5-6]。由此可知,多星应急调度具有如下特点:

1) 任务应尽可能在用户期望时间内完成,具有期望完成时间需求(定义为应急任务)。需要说明的是,期望完成时间并不是实时系统中的任务截止期,即使任务完成时间超出用户期望,任务依然执行。

2) 任务提交具有动态性,用户随时提交观测需求,并且一次可能提交大量任务,提交任务的数量和时间具有不确定性。

上述特点增加了应急条件下卫星资源管理调度的复杂性,因此,提出一种有效的应急条件多星动态调度方法具有重要意义。

目前,各国学者针对卫星调度问题开展了大量研究,这些研究主要是基于静态调度模式。Bensana 等[7]建立了约束满足模型和整数规划模型,提出了分支定界、Russian dolls 等确定性算法来求最优解,以及贪婪搜索、禁忌搜索等近似算法来获取可行解。Hall 和 Magazine[8]建立了近似于带时间窗口最大路径问题的单星调度模型,采用贪婪算法和动态规划技术求解模型。Lin 等[3,9-10]采用拉格朗日松弛技术,并结合禁忌搜

收稿日期: 2012-06-13; **退修日期**: 2012-09-14; **录用日期**: 2012-11-29; **网络出版时间**: 2013-01-09 11:15
网络出版地址: www.cnki.net/kcms/detail/11.1929.V.20130109.1115.007.html
基金项目: 国家自然科学基金(61104180, 71271216);国家"973"计划(6136101)
* **通讯作者**. Tel.: 0731-84574552 E-mail: xmzhu@nudt.edu.cn

引用格式: *Wang J J, Zhu X M, Wu C B, et al. Multi-satellite dynamic scheduling method for emergencies. Acta Aeronautica et Astronautica Sinica, 2013, 34(5): 1151-1164. 王建江,朱晓敏,吴朝波,等. 面向应急条件的多星动态调度方法. 航空学报, 2013, 34(5): 1151-1164.*

索、线性搜索来求解成像卫星静态调度问题。Globus 等[11-12]设计了进化算法，并将进化算法同爬山法、模拟退火、遗传等算法进行了比较。上述静态调度方法具有明确的调度周期，调度决策一旦制定，不能被修改，显然不适用于具有诸多不确定性的应急环境。

针对成像卫星动态调度，Pemberton 和 Greenwald[13]分析了其中存在的不确定性因素。Liao 和 Yang[14-15]提出了启发式算法 FAHA (Feasibility Adjustment Heuristic Algorithm)，根据实时更新的天气情况动态调整观测方案。Billups[16]研究了成像卫星动态调度算法，包括贪婪算法、遗传算法、基于整数规划的算法以及基于图论的算法等。Wang 等[17]提出求解多星动态调度问题的启发式算法，该算法基于任务迭代修复思想，采用启发式规则进行任务替换。以上动态调度方法均没有考虑任务完成时间需求，不能确保任务尽可能在用户期望时间内完成，因此不适用于应急环境。

观测任务合成能够促使多个相邻目标在同一视场内观测，有效提高卫星观测效率，受到学者们的普遍关注。文献[18]分析了合成约束条件，建立了任务合成图模型，提出了基于团划分思想的任务合成算法。但是该模型和算法仅适用于单星单轨条件下的任务合成问题，即每个任务最多只有一个时间窗口，对于多星调度问题并不适用。因此，本文考虑多个卫星且每个卫星存在多个轨道圈次的情况，建立了多星多轨任务合成图 (Multi-Satellite Multi-Orbit Task Merging Graph, MSMOTMG)模型，提出了多星多轨任务合成算法。

白保存等[19-20]建立了考虑任务合成的成像卫星调度模型，提出动态任务合成启发式(DTMH)算法和快速模拟退火(VFSA)算法对模型求解。但上述模型和算法都是基于静态调度模式提出的，且没有考虑任务完成时间需求，不适用于应急条件多星动态调度问题。因此，本文在考虑任务合成的基础上，结合任务修复、向后移位思想，提出了应急条件多星动态调度方法。

本文主要工作和创新点总结如下：

1) 针对具有用户期望完成时间的应急任务，建立了多星动态调度多目标数学规划模型。

2) 建立了 MSMOTMG 模型，在此基础上，提出了多星多轨应急任务合成算法 CP-TM。

3) 基于启发式规则，提出了综合考虑任务合成、修复和向后移位的多星动态应急调度(Dynamic Emergency Scheduling with Task Merging, Rehabilitation and Backward Shift, TMRBS-DES)算法。

1 应急条件动态调度模型

应急条件多星动态调度主要是针对具有期望完成时间、到达时间不确定的非周期任务。本节建立了多星动态调度多目标数学规划模型。

1.1 任务、资源和成像机会

本文定义的任务均为点目标任务(定义为元任务，Atom Task)。考虑元任务集合 $T=\{t_1,t_2,\cdots,t_{|T|}\}$，其中任意一个元任务可表示为 $t_i=(p_i,a_i,e_i)$，p_i、a_i 和 e_i 分别为 t_i 的优先级、到达时间和期望完成时间。结合实际，用户应对突发事件通常会一次提交大量任务，因此，本文假设元任务分批到达，每一批存在多个任务。

卫星资源集合 $R=\{r_1,r_2,\cdots,r_{|R|}\}$，其中任意一个资源可表示为 $r_j=(d_j,\sigma_j,s_j,b_j,o_j,\mathrm{as}_j,\mathrm{msg}_j)$，其中 d_j、σ_j、s_j、b_j、o_j、as_j 和 msg_j 分别为资源 r_j 的元任务执行时间、视场角、侧摆速率、启动时间、关机滞留时间、姿态稳定时间和最大侧摆角度。点目标能被传感器单个视场覆盖，大小可忽略不计，所以 r_j 上所有元任务执行时间相同，记为 d_j。

$\mathrm{AO}_{i,j}=\{\mathrm{ao}_{i,j,1},\mathrm{ao}_{i,j,2},\cdots,\mathrm{ao}_{i,j,K_{i,j}}\}$ 为元任务 t_i 在资源 r_j 上的成像机会集合，$K_{i,j}$ 为集合 $\mathrm{AO}_{i,j}$ 中元素个数，其中任意一个成像机会 $\mathrm{ao}_{i,j,k}\in\mathrm{AO}_{i,j}$ 可表示为 $\mathrm{ao}_{i,j,k}=\{[\mathrm{ws}_{i,j,k},\mathrm{we}_{i,j,k}],\theta_{i,j,k}\}$，其中 $[\mathrm{ws}_{i,j,k},\mathrm{we}_{i,j,k}]$ 为成像机会 $\mathrm{ao}_{i,j,k}$ 的时间窗口，$\theta_{i,j,k}$ 为理想侧摆角，如图 1 所示。

基于元任务 t_i 的期望完成时间，本文将成像机会集合 $\mathrm{AO}_{i,j}$ 划分为 3 个子集，即有效成像机会集合 $\mathrm{v\text{-}AO}_{i,j}$，临界成像机会集合 $\mathrm{c\text{-}AO}_{i,j}$ 和延迟成像机会集合 $\mathrm{d\text{-}AO}_{i,j}$。对于任意成像机会 $\mathrm{ao}_{i,j,k}$，1)如果 $\mathrm{we}_{i,j,k}\leqslant e_i$，则 $\mathrm{ao}_{i,j,k}\in\mathrm{v\text{-}AO}_{i,j}$；2)如果 $\mathrm{ws}_{i,j,k}\leqslant e_i<\mathrm{we}_{i,j,k}$，则 $\mathrm{ao}_{i,j,k}\in\mathrm{c\text{-}AO}_{i,j}$；3)如果 $\mathrm{ws}_{i,j,k}>e_i$，则 $\mathrm{ao}_{i,j,k}\in\mathrm{d\text{-}AO}_{i,j}$。集合 $\mathrm{v\text{-}AO}_{i,j}$、$\mathrm{c\text{-}AO}_{i,j}$ 和 $\mathrm{d\text{-}AO}_{i,j}$ 中元素个数分别为

$K^{\mathrm{v}}_{i,j}$、$K^{\mathrm{c}}_{i,j}$ 和 $K^{\mathrm{d}}_{i,j}$。显然,$K^{\mathrm{v}}_{i,j}$ 是一个有限整数,$K^{\mathrm{c}}_{i,j}$ 等于"1"或者"0",$K^{\mathrm{d}}_{i,j}$ 是一个无穷大的整数。由于存在系统内存、用户需求等约束,任务等待时间不宜过长,因此,本文假设每个元任务 t_i 具有有效完成时间 dd_i,并且元任务必须在其有效完成时间之前完成,所以对于不满足 $\mathrm{ws}_{i,j,k} < \mathrm{dd}_i$ 的成像机会本文不予考虑。因此,$K^{\mathrm{d}}_{i,j}$ 也定义为一个有限整数。显然存在 $K_{i,j} = K^{\mathrm{v}}_{i,j} + K^{\mathrm{c}}_{i,j} + K^{\mathrm{d}}_{i,j}$。对于所有资源,元任务 t_i 的成像机会数为 $\sum_{j=1}^{m} K_{i,j}$。

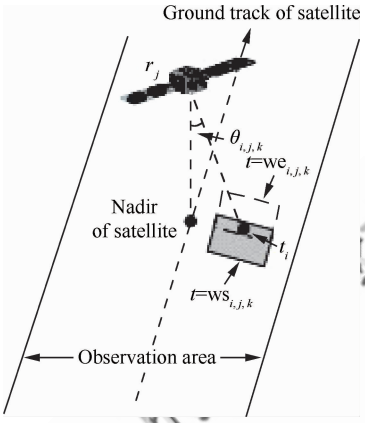


图 1 成像机会 $\mathrm{ao}_{i,j,k}$

Fig. 1 Imaging opportunity $\mathrm{ao}_{i,j,k}$

矩阵 $\boldsymbol{X} = [x_{i,j,k}]_{|T|\times|R|\times K_{i,j}}$ 表示任务调度信息,$x_{i,j,k} = 1$ 表示元任务 t_i 分配到资源 r_j 上的第 k 个成像机会执行,否则 $x_{i,j,k} = 0$。此外用 $\mathrm{bt}_{i,j}$、$\mathrm{ft}_{i,j}$ 和 $\phi_{i,j}$ 分别表示元任务 t_i 在资源 r_j 上的开始时间、结束时间和观测角度,且 $\mathrm{ft}_{i,j} = \mathrm{bt}_{i,j} + d_j$。

应急条件多星动态调度问题中存在 4 类任务:已完成任务(Finished Task, FT),正在执行任务(Executing Task, ET),等待任务(Waiting Task, WT)和新到达任务(New Task, NT),如图 2 所示,图中 t_{R} 为调度时刻。

任务分类与调度时刻 t_{R} 有关。对于元任务 t_i,1)如果 $\mathrm{ft}_{i,x} < t_{\mathrm{R}}$,则 $t_i \in \mathrm{FT}$;2)如果 $\mathrm{bt}_{i,x} \leqslant t_{\mathrm{R}} < \mathrm{ft}_{i,x}$,则 $t_i \in \mathrm{ET}$;3)如果 $\mathrm{bt}_{i,x} > t_{\mathrm{R}}$,则 $t_i \in \mathrm{WT}$;4)如果 $a_i = t_{\mathrm{R}}$,则 $t_i \in \mathrm{NT}$。由于本文采用非抢占方式,已完成任务和正在执行任务的调度决策不能改变,只有等待任务和新到达任务可以进一步优化。因此,本文调度主要针对 WT 和 NT 中的应急任务。

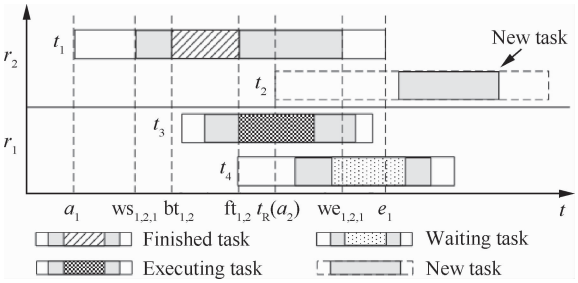


图 2 动态调度中的 4 类任务

Fig. 2 Four sorts of tasks in dynamic scheduling

1.2 调度约束

每个元任务只能分配到一个资源上,并且最多执行一次。因此,有如下约束:

$$C_1: \sum_{j=1}^{|R|}\sum_{k=1}^{K_{i,j}} x_{i,j,k} \leqslant 1 \quad \forall t_i \in T \tag{1}$$

元任务 t_i 必须在成像机会 $\mathrm{ao}_{i,j,k}$ 内执行,因此,有如下成像机会约束:

$$\forall t_i \in T, r_j \in R, \mathrm{ao}_{i,j,k} \in \mathrm{AO}_{i,j}, x_{i,j,k} = 1,$$
$$C_2: \begin{cases} \mathrm{bt}_{i,j} \in [\mathrm{ws}_{i,j,k}, \mathrm{we}_{i,j,k} - d_j] \\ \phi_{i,j} \in [\max(\theta_{i,j,k} - \sigma_j/2, -\mathrm{msg}_j), \\ \quad \min(\theta_{i,j,k} + \sigma_j/2, \mathrm{msg}_j)] \end{cases} \tag{2}$$

定义 1(合成任务(Composite Task)) 多个相邻的元任务,即 $T_{C_i} = \{t_{i1}, t_{i2}, \cdots, t_{iG}\}$,合成后产生的新任务称之为合成任务 t_{C_i}。

考虑 r_j 上等待执行的合成任务队列 $T_j = \{t_{C_1,j}, t_{C_2,j}, \cdots, t_{C_H,j}\}$($T_j \subset T \setminus \mathrm{FT}$,其中 $T \setminus \mathrm{FT}$ 表示 FT 的补集),即 $t_{C_1,j}$ 最先执行,$t_{C_H,j}$ 最后执行。为便于描述,T_j 中未经合成的元任务可看做只包含其自身的合成任务。

如果一个合成任务被成功执行,传感器需要充足的准备时间来执行下一任务。

定义 2(准备时间) 准备时间 $c_{i,i+1,j}$ 表示从合成任务 $t_{C_i,j}$ 执行结束到下一任务 $t_{C_{i+1},j}$ 开始执行需要的最小转换时间,其表达式为

$$c_{i,i+1,j} = o_j + |\phi_{C_{i+1},j} - \phi_{C_i,j}|/s_j + \mathrm{as}_j + b_j \tag{3}$$

式中:$\phi_{C_i,j}$ 和 $\phi_{C_{i+1},j}$ 分别为合成任务 $t_{C_i,j}$、$t_{C_{i+1},j}$ 的观测角度。

定义 3(准备就绪时间) 准备就绪时间定义为

$$\mathrm{rt}_{C_i,j} = \mathrm{ft}_{C_{i-1},j} + c_{i-1,i,j} \tag{4}$$

因此，准备就绪时间约束描述为

$$C_3: \forall t_{C_i,j} \in T_j, r_j \in R,\ \mathrm{rt}_{C_i,j} \leqslant \mathrm{bt}_{C_i,j} \tag{5}$$

合成任务中包含的元任务必须同时执行，因此，具有如下合成观测约束：

$$\forall r_j \in R, t_{C_i} \in T_j, t_{ig} \in t_{C_i}$$

$$C_4: \begin{cases} \sum_{k=1}^{K_{ig,j}} x_{ig,j,k} = 1, \mathrm{bt}_{ig,j} = \mathrm{bt}_{C_i,j} \\ \mathrm{ft}_{ig,j} = \mathrm{ft}_{C_i,j}, \phi_{ig,j} = \phi_{C_i,j} \end{cases} \tag{6}$$

1.3 调度目标

本文优先考虑调度收益，即最大化调度任务优先级：

$$\max\left\{\sum_{j=1}^{|R|}\sum_{i=1}^{|T|}\sum_{k=1}^{K_{i,j}} p_i x_{i,j,k}\right\} \tag{7}$$

然后，应急任务应尽量在用户期望时间内完成，即最大化期望完成时间内调度任务优先级：

$$\max\left\{\sum_{j=1}^{|R|}\sum_{i=1}^{|T|}\sum_{k=1}^{K^v_{i,j}} p_i x_{i,j,k}\right\} \tag{8}$$

最后，调度过程中应当最小化所有任务的扰动。在介绍该目标前，首先给出扰动的定义。

定义 4(扰动) 扰动 δ_α 是在第 α 次调度中，新生成的调度方案与原方案之间距离的度量。一般来说，距离主要由以下 3 类任务变化产生：

1) 任务完成时间在用户期望时间内改变。

2) 任务完成时间改变而不满足用户期望。

3) 任务被拒绝。

假设共有 s 批任务，由于调度是由一批新任务到达驱动的，所以总的调度次数为 s。因此

$$\delta = \sum_{\alpha=1}^{s}\delta_\alpha = \sum_{\alpha=1}^{s}\sum_{i=1}^{|T|}\sum_{\beta=1}^{3}\omega_\beta \mathrm{disturb}_\beta(i,\alpha) \tag{9}$$

式中：δ 为所有任务动态调度过程中总的扰动；$\omega_\beta(\beta = 1,2,3)$ 为任务第 β 类变化的影响因子，$\omega_1 < \omega_2 < \omega_3$；$\mathrm{disturb}_\beta(i,\alpha)$ 定义为

$$\mathrm{disturb}_\beta(i,\alpha) = \begin{cases} 1, \text{在第}\alpha\text{次调度中任务} \\ \quad t_i \text{发生第}\beta\text{类变化} \\ 0, \text{其他} \end{cases} \tag{10}$$

因此，最小扰动目标为

$$\min\sum_{\alpha=1}^{s}\sum_{i=1}^{|T|}\sum_{\beta=1}^{3}\omega_\beta \mathrm{disturb}_\beta(i,\alpha) \tag{11}$$

2 任务合成

为了提高卫星观测效率，减少任务冲突，本节提出了应急条件下多星多轨任务合成策略。

2.1 合成约束

假设有 G 个元任务 $T_{C_i} = \{t_{i1}, t_{i2}, \cdots, t_{iG}\}$ 能够合成，生成一个合成任务 t_{C_i}，其合成约束如下所示：

1) 可用资源约束

$$\exists r_j \in R: \forall t_{ig} \in T_{C_i}, K_{ig,j} > 0 \tag{12}$$

2) 成像机会约束

$$\forall t_{ig} \in T_{C_i}, \exists r_j \in R, \mathrm{ao}_{ig,j,k_{ig}} \in \mathrm{AO}_{ig,j}$$

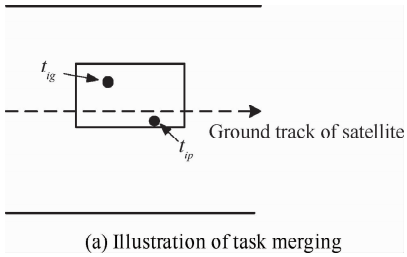
$$\text{s.t.} \begin{cases} \text{(a)}: \max\limits_{t_{ig}\in T_{C_i}}\{\theta_{ig,j,k_{ig}}\} - \min\limits_{t_{ig}\in T_{C_i}}\{\theta_{ig,j,k_{ig}}\} \leqslant \sigma_j \\ \text{(b)}: \min\limits_{t_{ig}\in T_{C_i}}\{\mathrm{we}_{ig,j,k_{ig}}\} - \max\limits_{t_{ig}\in T_{C_i}}\{\mathrm{ws}_{ig,j,k_{ig}}\} \geqslant d_j \end{cases} \tag{13}$$

可用资源约束表示存在资源 r_j 能够执行 T_{C_i} 中的所有元任务，即对所有元任务都有成像机会。成像机会约束表示 T_{C_i} 中所有元任务都能够在资源 r_j 的一个视场内观测，如图 3(a)所示。

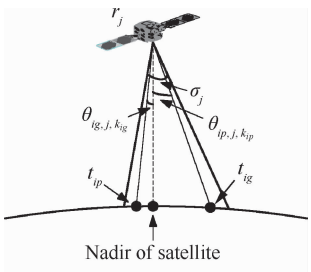
由合成任务成像机会约束可得：

1) 式(13)中(a)为角度约束。通过调整观测角度，传感器视场能够覆盖所有点目标，如图 3(b)所示。

2) 式(13)中(b)为时间约束。T_{C_i} 中所有元任务的时间窗口相交，即存在对所有目标均可成像的时段，且该成像时段长度不小于元任务执行时间，如图 3(c)所示。



(a) Illustration of task merging



(b) Angle constraint of merging

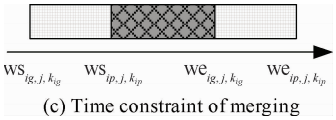


图 3 任务合成约束

Fig. 3 Task merging constraints

由以上分析可知,对于任意一个元任务 $t_{ig} \in T_{C_i}$,存在成像机会 $ao_{ig,j,k_{ig}}$ 满足约束式(13)。因此 T_{C_i} 中各个元任务满足合成约束的成像机会构成集合 $MO_{C_i,j,k}=\{ao_{i1,j,k_{i1}},ao_{i2,j,k_{i2}},\cdots,ao_{iG,j,k_{iG}}\}$,称为合成机会(Merging Opportunity)。类似地,多个任务 $T_{C_i}=\{t_{i1},t_{i2},\cdots,t_{iG}\}$ 间可能存在多个合成机会。对于每个合成机会 $MO_{C_i,j,k}$,合成任务 t_{C_i} 具有一个成像机会 $ao_{C_i,j,k} \in AO_{C_i,j}$。图 4 描述了一个成像机会合成的示例,元任务 t_{i1}、t_{i2} 和 t_{i3} 能够构成一个合成任务 t_{C_i}。元任务在资源 r_j 上的成像机会 $ao_{i1,j,2}$、$ao_{i2,j,1}$ 和 $ao_{i3,j,1}$ 构成了一个合成机会 $MO_{C_i,j,1}$。类似地,$ao_{i1,j,3}$、$ao_{i2,j,3}$ 和 $ao_{i3,j,2}$ 构成了合成机会 $MO_{C_i,j,2}$。$MO_{C_i,j,1}$ 和 $MO_{C_i,j,2}$ 分别对应于合成任务 t_{C_i} 的成像机会 $ao_{C_i,j,1}$ 和 $ao_{C_i,j,2}$。因此,合成任务 t_{C_i} 在资源 r_j 上的成像机会集合为 $AO_{C_i,j}=\{ao_{C_i,j,1},ao_{C_i,j,2}\}$。

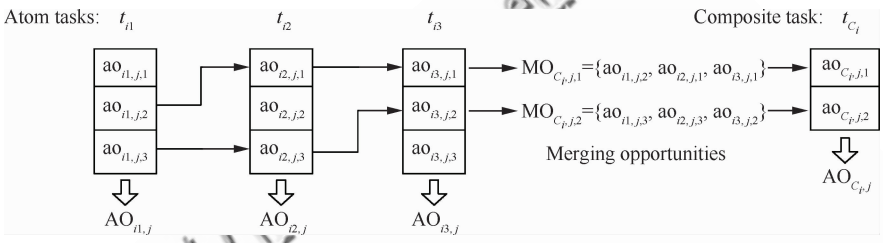


图 4 合成任务成像机会

Fig. 4 Imaging opportunities for a composite task

成像机会 $ao_{C_i,j,k}$ 的理想侧摆角 $\theta_{C_i,j,k}=(\max\limits_{t_{ig}\in T_{C_i}}(\theta_{ig,j,k_{ig}})+\min\limits_{t_{ig}\in T_{C_i}}(\theta_{ig,j,k_{ig}}))/2$,时间窗口为 $[ws_{C_i,j,k},we_{C_i,j,k}]$,$ws_{C_i,j,k}=\max\limits_{t_{ig}\in T_{C_i}}(ws_{ig,j,k_{ig}})$,$we_{C_i,j,k}=\min\limits_{t_{ig}\in T_{C_i}}(we_{ig,j,k_{ig}})$。$p_{C_i}$ 和 e_{C_i} 分别为合成任务 t_{C_i} 的优先级和期望完成时间,其中,$p_{C_i}=\sum\limits_{g=1}^{G}p_{ig}$,$e_{C_i}=\min\limits_{t_{ig}\in T_{C_i}}(e_{ig})$。

2.2 模型与算法

本文提出 MSMOTMG 模型,在合成图 G 中,每个顶点表示一个合成任务,所有顶点构成顶点集 $V(G)$。元任务可看做只包含其自身的合成任务。对于任意两个顶点 t_{C_i} 和 t_{C_j},如果 t_{C_i} 和 t_{C_j} 中包含的元任务 $t_{P1},t_{P2},\cdots,t_{PG}$ 满足合成约束,能够构成一个新的合成任务 t_{C_P},那么 t_{C_i} 和 t_{C_j} 存在边连接关系。每条边 $e_{C_P,j,k}$ 与合成机会 $MO_{C_P,j,k}=\{ao_{P1,j,k_{P1}},ao_{P2,j,k_{P2}},\cdots,ao_{PG,j,k_{PG}}\}$ 一一映射,可由下面元组表示:

$$e_{C_P,j,k}=(r_j,k_{P1},k_{P2},\cdots,k_{PG})$$

t_{C_i} 和 t_{C_j} 之间所有的边构成集合 $E_{i,j}$,$E(G)$ 为图 G 中边集的集合。

图 5 是一个包含 5 个合成任务的 MSMOTMG 模型,其中 $V(G)=\{t_{C_1},t_{C_2},t_{C_3},t_{C_4},t_{C_5}\}$,$t_{C_2}$ 包含两个元任务 t_2 和 t_3,其他合成任务仅包含一个元任务。合成图 G 中边集的集合为 $E(G)=\{E_{1,2},E_{1,3},E_{2,3},E_{2,4}\}$,其中 $E_{1,2}=\{e_{C_{P1},1,1},e_{C_{P2},2,1}\}$,$E_{1,3}=\{e_{C_{P2},1,1},e_{C_{P2},2,1}\}$,$E_{2,3}=\{e_{C_{P3},1,1}\}$,$E_{2,4}=\{e_{C_{P4},2,1}\}$。

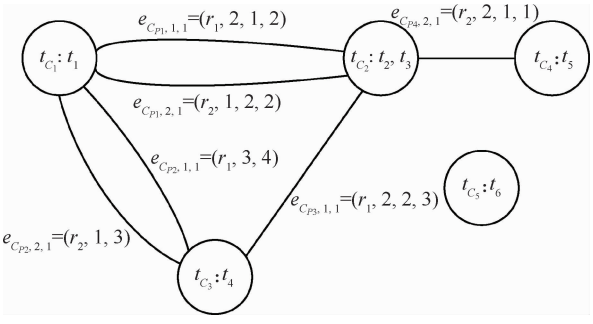


图 5 多星多轨任务合成图 G

Fig. 5 Multi-satellite multi-orbit task merging graph G

定理 1 如果集合 $E_{i,j}$ 中存在 m 个元素，由 t_{C_i} 和 t_{C_j} 构成的合成任务 t_{C_P} 具有 m 个成像机会。

证明 假设 t_{C_i} 和 t_{C_j} 中的元任务为 $t_{P1}, t_{P2}, \cdots, t_{PG}$。由 2.1 节可知，合成机会 $\mathrm{MO}_{C_P,j,k} = \{\mathrm{ao}_{P1,j,k_{P1}}, \mathrm{ao}_{P2,j,k_{P2}}, \cdots, \mathrm{ao}_{PG,j,k_{PG}}\}$ 与合成任务 t_{C_P} 的成像机会 $\mathrm{ao}_{C_P,j,k}$ 一一映射。同时，由任务合成图可知元素 $e_{C_P,j,k}$ 与合成机会 $\mathrm{MO}_{C_P,j,k}$ 一一映射。

因此，合成任务 t_{C_P} 的成像机会 $\mathrm{ao}_{C_P,j,k}$ 与集合 $E_{i,j}$ 中的元素 $e_{C_P,j,k}$ 一一映射。由此可知，如果集合 $E_{i,j}$ 中有 m 个元素，则合成任务 t_{C_P} 有 m 个成像机会。

称 $\mathrm{ao}_{C_P,j,k}$ 为合成任务 t_{C_P} 的有效成像机会，如果 $\min\limits_{t_{Pi}\in t_{C_P}}(\mathrm{we}_{P_i,j,k_{Pi}}) \leqslant \min\limits_{t_{Pi}\in t_{C_P}}(e_{P_i})$，对应的元素 $e_{C_P,j,k}$ 称为有效元素，其中，e_{P_i} 为任务 t_{Pi} 的期望完成时间。由上述定理得，如果集合 $E_{i,j}$ 中元素越多，合成任务 t_{C_P} 的成像机会就越多，调度成功率随之提高。此外，合成任务应当具有较多的有效成像机会，确保任务尽可能在期望完成时间内执行。因此，任务合成优先级定义如下：

定义 5(合成优先级(Merging Priority, MP)) 合成优先级定义为

$$\mathrm{MP}(E_{i,j}) = |E_{i,j}| + |E_{i,j}|_{\mathrm{valid}} \tag{14}$$

式中：$|E_{i,j}|$ 和 $|E_{i,j}|_{\mathrm{valid}}$ 分别为集合 $E_{i,j}$ 中元素和有效元素的个数。

本文提出多星多轨应急任务合成算法 CP-TM(见图 6)，描述如下：

CP-TM 算法首先合成具有最大合成优先级的任务 t_{C_i} 和 t_{C_j}，产生一个新的合成任务 t_{C_P}，更新图 G。然后，算法选择下一组具有最大合成优先级的任务进行合成，直到集合 $E(G)$ 为空，算法结束。

Algorithm 1. CP-TM algorithm

1. Establish MSMOTMG G for the whole tasks in NT, calculate $\mathrm{MP}(E_{i,j})$ for each set $E_{i,j}$ in G;
2. **while** $E(G) \neq \varnothing$
3. Select edge set $E_{i,j}$ with maximum $\mathrm{MP}(E_{i,j})$;
4. **for** each vertex t_{C_k} in G except t_{C_i}, t_{C_j}
5. Delete the edge sets $E_{i,k}$ and $E_{j,k}$;
6. **end for**
7. Merge verticles t_{C_i}, t_{C_j} to generate a new vertex t_{C_P};
8. Delete t_{C_i}, t_{C_j};
9. **for** each vertex t_{C_k} in G except t_{C_P}
10. Generate edge set $E_{P,k}$;
11. **end for**
12. **end while**

图 6 CP-TM 算法

Fig. 6 CP-TM algorithm

任务合成可能会减少任务成像机会，导致调度成功率降低。为克服该缺陷，本文提出一种基于合成任务分解的修复技术。如果合成任务 t_{C_P} 调度失败，取出其中优先级最大的元任务 t_{Pi}，然后其他元任务构成一个新的合成任务 $t_{C_P\setminus Pi}$，合成任务 t_{C_P} 被分解为具有更多成像机会的合成任务 $t_{C_P\setminus Pi}$ 和元任务 t_{Pi}，把它们插入 NT 中重新调度。

定理 2 CP-TM 算法时间复杂度为 $O(n^3)$，其中 n 为新任务集 NT 中的任务数。

证明 建立 MSMOTMG 的时间复杂度为 $O(n^2)$(Line 1)。具有 n 个顶点的任务合成图最多有 C_n^2 个边集，则选择具有最高合成优先级的边集需要消耗的时间为 C_n^2 (Line 3)，所以算法中一次 while 循环消耗的时间为 $C_n^2+n+1+1+n$ (Lines 3～11)，其时间复杂度为 $O(n^2)$。算法中每循环一次做一次任务合成，任务数量减一，则最多循环 $n-1$ 次，算法终止。所以，CP-TM 算法时间复杂度为 $O(n^2)+O(n^3)$，即 $O(n^3)$。

3 应急条件动态调度算法

多星动态调度是 NP-complete 问题[13,17]，本文采用启发式算法求解近似最优解。

3.1 任务插入

任务插入是指在满足相关约束的基础上，将 NT 中的任务逐个插入到等待任务集 WT 中。新任务通常插入到 $T_j(j \in \{1,2,\cdots,|R|\})$ 中相邻任务间的空闲时间片上。特例是任务插入到第 1 个任务前或者最后一个任务后。不失一般性，在每个队列 T_j 中增加两个虚拟任务，即初始任务 $t_{s,j}$ 和终结任务 $t_{e,j}$，它们具有如下性质：

1) 任务 $t_{s,j}$ 的开始时间为调度时刻 t_R，结束时间 $\mathrm{ft}_{s,j}$ 为 t_R 时刻正在执行任务的结束时间。如果没有任务正在执行，则 $\mathrm{ft}_{s,j} = t_R$。

2) 任务 $t_{e,j}$ 的开始时间和结束时间均为 ∞。

3) 任务 $t_{s,j}$、$t_{e,j}$ 与其他任务的准备时间为 0。

基于上述分析，任务插入可统一描述为考虑等待任务向后移位，将任务插入到 $t_{C_k,j}$ 和 $t_{C_{k+1},j}$ 之间的空闲时间片，记为 $< r_j, t_{C_k,j}, t_{C_{k+1},j} >$。

定义 6(后移空余时间)[21] 后移空余时间

$B_i^{(i,i+1,\cdots,j)}$ 表示在保证队列 $(t_{C_i,j},t_{C_{i+1},j},\cdots,t_{C_j,j})$ 中任务执行的条件下,任务 $t_{C_i,j}$ 可向后移动的时间范围。如果省略 $B_i^{(i,i+1,\cdots,j)}$ 的上标,即 B_i ,表示队列为从 $t_{C_i,j}$ 到 T_j 中的最后一个任务。

计算 B_i 的细节见文献[21]。将任务 t_{C_i} 插入到时间片 $< r_j,t_{C_k,j},t_{C_{k+1},j} >$ 的约束条件如下所示。

可用资源约束:

$$K_{C_i,j} > 0 \tag{15}$$

时间约束:

$$\exists k_i \in \{1,2,\cdots,K_{C_i,j}\}$$
$$\text{s.t.}\begin{cases}\text{(a)}:\ \mathrm{rt}_{C_i,j} \leqslant \mathrm{we}_{C_i,j,k_i} - d_j \\ \text{(b)}:\ \mathrm{rt}_{C_{k+1},j} \leqslant \mathrm{bt}_{C_{k+1},j} + B_{k+1}\end{cases} \tag{16}$$

可用资源约束表示资源 r_j 能够执行任务 t_{C_i} ,即 t_{C_i} 在 r_j 上有成像机会。对于时间约束:

1) 式(16)中(a)表示任务 t_{C_i} 的开始时间必须满足成像机会约束(见式 2)。

2) 式(16)中(b)表示任务 t_{C_i} 的插入必须确保后续任务的顺利执行。

如果任务 t_{C_i} 满足约束式(15)~式(16),表示其可以被插入到时间片 $< r_j,t_{C_k,j},t_{C_{k+1},j} >$,得到一个新的等待任务 $t_{C_i,j}$,新任务开始时间为 $\mathrm{bt}_{C_i,j} = \max(\mathrm{rt}_{C_i,j},\mathrm{ws}_{C_i,j,k_i})$,结束时间为 $\mathrm{ft}_{C_i,j} = \mathrm{bt}_{C_i,j} + d_j$,观测角度为 $\phi_{C_i,j} = \theta_{C_i,j,k_i}$ 。

任务 t_{C_i} 的所有可插入时间片构成可选时间片(Alternative Time Slot)集合 ATSS_i。本文可选时间片构建(ATSE)算法就是用来构造集合 ATSS_i,具体算法流程如图 7 所示。

定理 3 ATSE 算法的时间复杂度为 $O(n^2m)$,其中 n 为调度任务总数,m 为单个任务最大成像机会数。

证明 重新计算任务准备就绪时间需要消耗的时间为 $O(n)$ (Lines 10~12),更新后续任务执行时间为 $O(n)$ (Lines 14~21)。因此,ATSE 算法的时间复杂度为 $O(m)O(n)O(n+n) = O(n^2m)$ 。

图 7 中:n_p 为任务 t_{C_i} 插入到时间片 $< r_j,t_{C_k,j},t_{C_{k+1},j} >$ 后推迟的任务数量;n_v 为任务插入后由于向后移位导致的不满足用户期望时间的任务数量。上述两个参数对于选择任务插入的时间片至关重要,将在 3.2 节中使用。

Algorithm 2. ATSE algorithm

1. Initialize the set $\mathrm{ATSS}_i=\varnothing$ imaging opportunity set $\mathrm{AO}_{C_i}=\bigcup_{r_j\in R}\mathrm{AO}_{C_i,j}$;
2. **for** each imaging opportunity ao_{C_i,j,k_i} in AO_{C_i}
3. **for** each task $t_{C_k,j}$ in T_j
4. **if** ao_{C_i,j,k_i} satisfies the constraints Eqs.(15)-(16) with $< r_j,t_{C_k,j},t_{C_{k+1},j} >$
5. Set $n_\mathrm{p}=0,n_\mathrm{v}=0$;
6. Calculate $\mathrm{rt}_{C_i,j}$, $\mathrm{bt}_{C_i,j}$ and $\mathrm{ft}_{C_i,j}$, assume task t_{C_i} is inserted to $< r_j,t_{C_k,j},t_{C_{k+1},j} >$;
7. **if** $\mathrm{ft}_{C_i,j} > \mathrm{et}_{C_i}$
8. $n_\mathrm{v} = n_\mathrm{v}+1$
9. **end if**
10. **for** each task $t_{C_j,j}$ after $t_{C_i,j}$ in T_j
11. Recalculate $\mathrm{rt}'_{C_j,j}$;
12. **end for**
13. $t_{C_j,j} \leftarrow t_{C_{k+1},j}$;
14. **while** $\mathrm{rt}'_{C_j,j} > \mathrm{bt}_{C_j,j}$
15. Recalculate $\mathrm{bt}'_{C_j,j}$, $\mathrm{ft}'_{C_j,j}$;
16. $n_\mathrm{p} = n_\mathrm{p}+1$;
17. **if** $\mathrm{ft}_{C_j,j} \le \mathrm{et}_{C_j}$ and $\mathrm{ft}'_{C_j,j} > \mathrm{et}_{C_j}$
18. $n_\mathrm{v} = n_\mathrm{v}+1$;
19. **end if**
20. $t_{C_j,j} \leftarrow t_{C_{j+1},j}$;
21. **end while**
22. $\mathrm{ATSS}_i = \mathrm{ATSS}_i \cup < r_j,t_{C_k,j},t_{C_{k+1},j} >$;
23. **end if**
24. **end for**
25. **end for**

图 7 ATSE 算法

Fig. 7 ATSE algorithm

3.2 规 则

应急条件下成像卫星动态调度中存在两个关键步骤,即选择新任务和选择任务插入的时间片。为了解决该问题,本文参考了任务需求度[18]规则,并设计了任务优先插入规则。

3.2.1 任务需求度

任务需求度 TRM_{C_i} 表示了任务 t_{C_i} 需要调度的紧迫程度,其表达式为

$$\mathrm{TRM}_{C_i} = \frac{p_{C_i}}{\sum_{r_j\in R} K_{C_i,j}} \tag{17}$$

因此,具有较高优先级和较少成像机会的任务应当优先调度。该规则适用于在 NT 中选择新任务。

3.2.2 任务优先插入规则

具有最小 P 值的时间片为任务的最佳插入时间片,其定义为

$$P = n_p + n_v^2 \tag{18}$$

n_p 增加会导致扰动增加，n_v 增加不仅增大调度扰动，同时会减少期望时间内的调度任务优先级(见1.3节中相关定义)。因此，应当在调度过程中尽量减小 n_p 和 n_v，尤其是 n_v 的值，即选择 $ATSS_i$ 中具有最小 P 值的时间片插入任务。

3.3 算法描述

本节提出了综合考虑任务合成、修复和向后移位的多星动态应急调度(TMRBS-DES)算法，算法描述如图8所示。

Algorithm 3. TMRBS-DES algorithm
1. Initialize the set T and compute $AO_{i,j}$ for each task t_i and resource r_j;
2. Merge the tasks in NT with **CP-TM algorithm**;
3. **while** NT $\neq \varnothing$
4. Select task t_{C_i} in NT with maximum TRM_{C_i}, NT = NT $\setminus t_{C_i}$;
5. Establish $ATSS_i$ for t_{C_i} with **ATSE algorithm;**
6. **if** $ATSS_i \neq \varnothing$
7. Select the time slot $< r_j, t_{C_k,j}, t_{C_{k+1},j} >$ in $ATSS_i$ with minimum P;
8. Insert the task t_{C_i} to $< r_j, t_{C_k,j}, t_{C_{k+1},j} >$;
9. **for** each task $t_{C_j,j}$ after t_{C_i} in T_j
10. Update $rt_{C_j,j}$, $bt_{C_j,j}$ and $ft_{C_j,j}$;
11. **end for**
12. break;
13. **end if**
14. **if** t_{C_i} is an atom task
15. Reject t_{C_i};
16. **else**
17. Select a task with maximum priority in t_{C_i}, say t_{Pi}, divide the composite task t_{C_i} into two tasks: t_{Pi} and $t_{C_i \setminus Pi}$, NT = NT $\bigcup t_{Pi}$, NT = NT $\bigcup t_{C_i \setminus Pi}$;
18. **end if**
19. **end while**

图8 TMRBS-DES算法

Fig. 8 TMRBS-DES algorithm

定理4 TMRBS-DES算法的时间复杂度为 $O(n^3 s^3 m)$，其中，n 为单批次最大任务数，s 为调度任务批次总数，m 为单个任务最大成像机会数。

证明 由定理2和定理3可知，CP-TM算法时间复杂度为 $O(n^3)$，ATSE算法时间复杂度为 $O(n^2 s^2 m)$。选择需求度最大的任务需要消耗的时间为 $O(n)$(Line 4)，而选择可选时间片插入任务的时间复杂度为 $O(m+ns)$(Lines 6～13)，所以，TMRBS-DES调度一批新任务的时间复杂度为 $O(n^3)+O(n)(O(n)+O(n^2 s^2 m)+O(m+ns))=O(n^3 s^2 m)$。因此，TMRBS-DES算法调度 s 批任务的时间复杂度为 $O(n^3 s^3 m)$。

4 模拟测试结果与分析

本文通过大量模拟实验测试TMRBS-DES算法的性能，将其与文献[17]中的RBHA算法，以及其他baseline算法进行比较，这些baseline算法为：只考虑任务向后移位的动态应急调度(BS-DES)算法，只考虑任务合成与修复的动态应急调度(TMR-DES)算法，以及不考虑任务修复的动态应急调度(TMBS-DES)算法。

1) RBHA：该算法的基本思想是在新任务的成像机会中进行迭代修复。当新任务 $t_i \in$ NT 与方案中的等待任务发生冲突时，基于规则选择任务退出当前方案以便新方案接受 t_i。

2) BS-DES：该算法不考虑任务合成与修复，仅考虑任务在等待队列中的向后移位。

3) TMR-DES：该算法只考虑任务合成与修复，任务在等待队列中不能向后移位。

4) TMBS-DES：该算法不考虑任务修复，当合成任务调度失败时，被系统直接丢弃。

为了比较的公平性，本文对RBHA算法稍做修改，首先在 $\bigcup_{r_j \in R}$ v-$AO_{i,j}$ 的成像机会中进行迭代修复搜索。如果搜索失败，再在其他成像机会中进行搜索。

系统评估指标包括：

1) 调度任务优先级(Total Scheduled Task Priorities, TSTP)，定义为 $TSTP = \sum_{j=1}^{|R|} \sum_{i=1}^{|T|} \sum_{k=1}^{K_{i,j}} p_i x_{i,j,k}$。

2) 期望完成时间内调度任务优先级(Total Scheduled Task Priorities within Expected Finish Time, TSTPEFT)，定义为 $TSTPEFT = \sum_{j=1}^{|R|} \sum_{i=1}^{|T|} \sum_{k=1}^{K^v_{i,j}} p_i x_{i,j,k}$。

3) 扰动测度(Perturbation Measurement)。

4.1 测试方法与参数

为了验证TMRBS-DES算法的性能，目标在纬度 $-30°$～$60°$，经度 $0°$～$150°$ 的区域中均匀随机分布。目标规模分别为200、400、600、800、1 000和1 200。不失一般性，任务优先级在区间[1,10]中均匀分布。本文考虑了在不同卫星平台

上的 3 个传感器资源。表 1 给出了传感器相关参数,其中卫星轨道模型来自卫星仿真工具 STK,带 * 号的参数是笔者根据文献拟定的,其他参数为卫星和传感器的真实数据。

1) 任务批次到达时间可表示为 $a_i = a_{i-1} +$ IntervalTime,其中 IntervalTime 为非负的均匀分布随机数。

2) batchSize 表示每一批次任务数量,是一个随机正整数,服从均匀分布。

3) t_i 的期望完成时间为 $e_i = a_i +$ Time,其中,Time$\sim N$(baseTime,baseTime/10)。

4) 任务 t_i 的有效完成时间为 $dd_i = a_i +$ DueDate,其中 DueDate 为正态分布随机数,DueDate$\sim N$(baseDueDate,baseDueDate/10)。

表 1 卫星传感器参数

Table 1 Parameters of satellite sensors

Sensor	Satellite	msg/(°)	σ/(°)	d^*/s	s^*/((°)·s^{-1})	b^*/s	o^*/s	as^*/s
Sensor 1	IKONOS-2	45	0.931	2	1	3	3	5
Sensor 2	QuickBird-2	25	2.1	2	1	3	3	5
Sensor 3	SPOT-5	27	2.09	2	1	3	3	5

表 2 列出了仿真相关参数及其取值范围,其中,有效完成时间默认值为 24 h,在此期间每个卫星大概运行十余个轨道圈次,因此,本实验模拟了多星多轨任务合成与调度问题,每个任务在同一资源上可能存在多个成像机会。此外,1.3 节中参数 ω_1、ω_2 和 ω_3 分别取值为 0.5、1 和 2。

表 2 仿真参数

Table 2 Parameters for simulation

Parameter	Value (Fixed)-(Varied)
Task number	(800)-(200,400,600,800,1 000,1 200)
Resource number	(3)
IntervalTime/h	([0,4])-([0,4],[4,8],[8,12])
batchSize	([200,300])
baseTime/h	(6)-(3, 6, 9, 12)
baseDueDate/h	(24)-(12, 24, 36, 48)

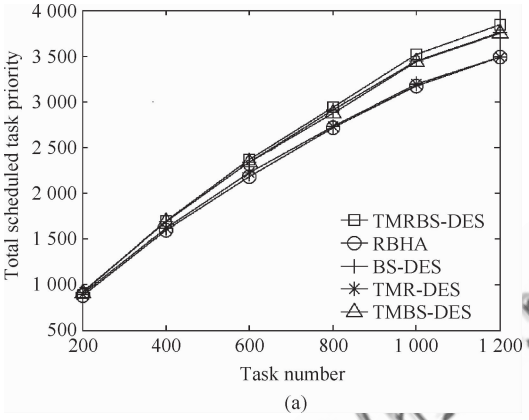
4.2 任务规模对算法的影响

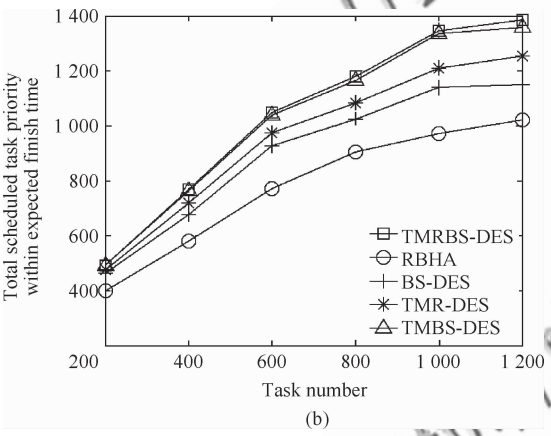
本组实验考察任务规模对算法性能的影响。任务数量从 200 到 1 200 不断递增。图 9 分别从调度任务优先级、期望时间内调度任务优先级、扰动测度等指标评估了算法 TMRBS-DES、RBHA、BS-DES、TMR-DES 和 TMBS-DES 的性能。

图 9(a)表示随着任务规模的增加,所有算法调度任务的优先级增加。这是由于资源能力充足,系统调度任务数量随着到达任务数量增大而增加。此外,TMRBS-DES 算法调度任务优先级最高,这是因为 TMRBS-DES 算法综合考虑了任务合成、修复和向后移位,提高了调度质量。相反,RBHA 算法调度任务优先级之和最小,这是因为 RBHA 算法不考虑任务合成与修复,且任务在等待队列中不能向后移位,增加了任务间的冲突。由于任务合成能够使多个任务同时观测,任务间的冲突机率随着任务规模减小而降低,所以 TMRBS-DES 算法性能优于 BS-DES 算法。TMBS-DES 算法没有考虑任务修复,使得合成任务由于成像机会减少而调度失败,所以其调度收益低于 TMRBS-DES 算法。TMRBS-DES 算法的性能优于 TMR-DES 算法,验证了向后移位能够灵活调整任务开始时间,减少任务冲突,提高调度成功率。

任务规模增大,用户期望时间内调度任务优先级随之增加,如图 9(b)所示。并且随着任务规模不断增加,用户期望时间内调度任务优先级增长率不断减小,这是由于用户期望时间一定,其可调度任务数量随着任务规模增加不断趋于饱和。TMRBS-DES 算法和 TMBS-DES 算法期望时间内的调度任务优先级基本相同,表明任务修复对期望时间内调度任务数量影响极小。此外,由于 RBHA 算法不考虑任务合成和向后移位,难以消解任务间存在的大量时间冲突,使得很多任务调度不能满足用户期望,所以其期望时间内调度任

务优先级最小。TMRBS-DES 算法期望时间内调度任务优先级略高于 TMR-DES 算法，表明任务向后移位增加了在用户期望时间内的完成任务数量。由于不考虑任务合成，任务间的大量冲突使得大多数任务调度不满足用户期望，所以 BS-DES 算法期望时间内调度任务优先级小于 TMRBS-DES 算法。





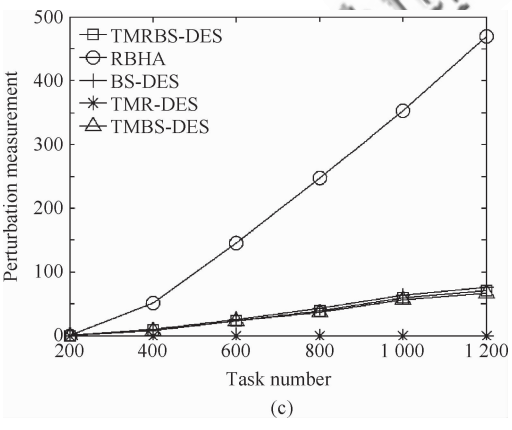


图 9 任务规模对算法性能的影响

Fig. 9 Algorithm performance impact of task number

由图 9(c)可以看出随着任务规模增加，所有算法扰动测度增大。算法 TMRBS-DES、BS-DES 和 TMBS-DES 仅仅考虑了任务向后移位，而不能取消已安排任务，扰动测度基本相等。相反，RBHA 算法采用了任务取消策略，已安排任务有可能在重调度中被拒绝，所以其扰动测度最大。TMR-DES 算法既不进行任务移位，也不考虑任务取消，已安排任务不可能发生变化，所以扰动测度始终为 0。

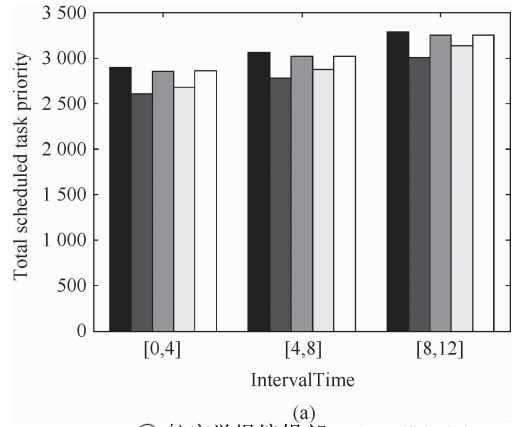
4.3 任务到达率对算法的影响

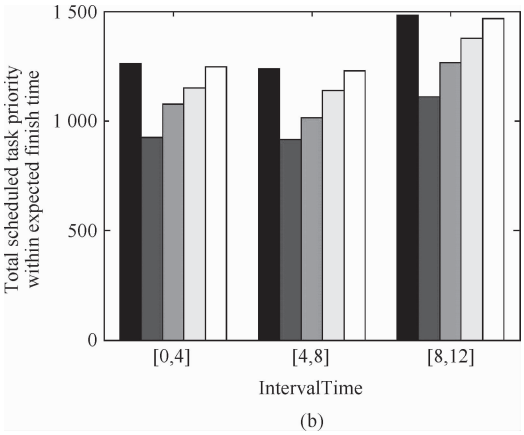
本组实验评估任务到达率对算法性能的影响。参数 IntervalTime 分别在区间[0，4]，[4，8]，[8，12]间均匀选取，实验结果如图 10 所示。

图 10(a)表明随着 IntervalTime 增大，所有算法调度任务的优先级增加。这是由于 IntervalTime 增大，则任务到达率减小，导致每个资源上的等待任务减少，任务间的冲突数随之减少，因此，系统能够接受更多的任务。

由图 10(b)可知，由于任务期望完成时间长度是一定的，期望完成时间内调度任务数量基本不变，所以用户期望时间内调度任务优先级基本不变。然而，随着 IntervalTime 进一步增大，用户期望时间内调度任务数量随之增加，因此期望时间内调度任务优先级又呈现上升趋势。

由于采用任务取消策略，导致更多等待任务在重调度中发生变化，甚至被系统拒绝，因此 RBHA 算法扰动测度最大，如图 10(c)所示。任务到达率减小，系统负载随之减轻，在重调度中需要改变的任务数量减少，所以算法扰动测度下降。





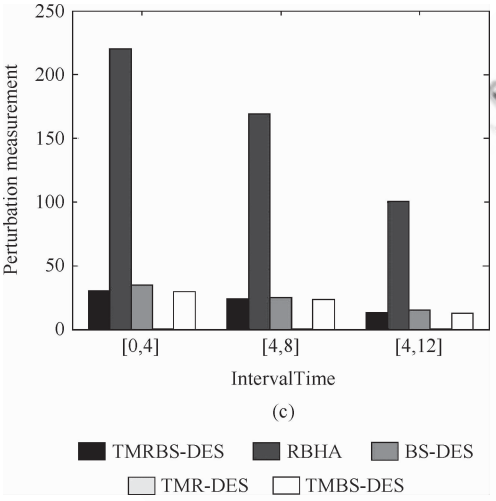


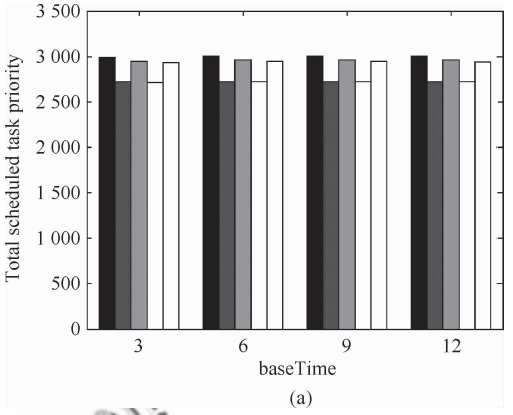
图 10 任务到达率对算法性能的影响

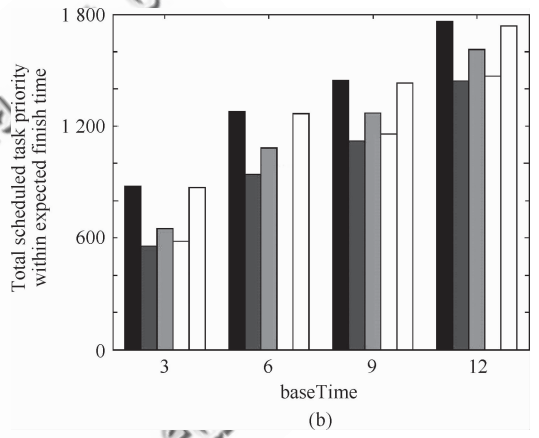
Fig. 10 Algorithm performance impact of arrival rate

4.4 期望完成时间对算法的影响

本组实验评估期望完成时间对算法的影响,实验结果如图 11 所示。由图 11(a)可知随着期望完成时间延长,算法调度任务优先级基本不变,这是因为任务有效完成时间不变,算法调度任务总数基本不变。任务期望完成时间延长促使用户期望时间内调度任务数量增加,因此期望完成时间内调度任务优先级随着任务期望完成时间增加而增大,如图 11(b)所示。

由图 11(c)可得,随着期望完成时间增加,系统扰动测度基本不变,说明期望完成时间的延长对调度扰动基本无影响。





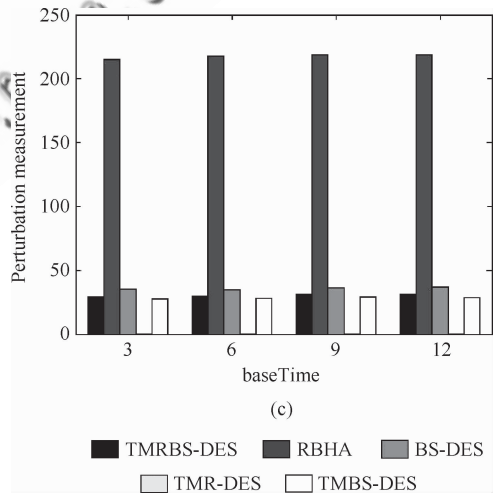


图 11 期望完成时间对算法性能的影响

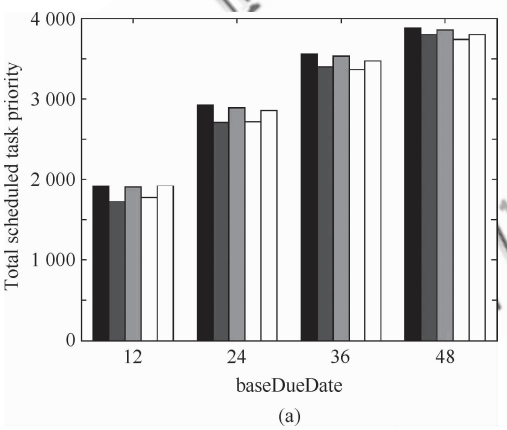
Fig. 11 Algorithm performance impact of expected finish time

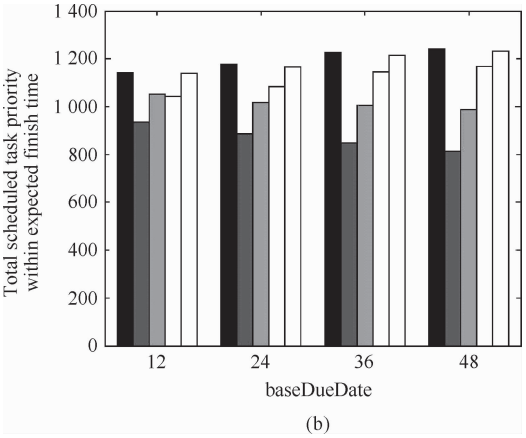
4.5 有效完成时间对算法的影响

本组实验评估任务有效完成时间对算法的影响,实验结果如图 12 所示。

图 12(a)表明任务有效完成时间延长，系统调度任务优先级增大，原因是任务有效完成时间延长，任务成像机会增加，提高了调度成功率。如图 12(b)所示，随着 baseDueDate 增大，TMRBS-DES、TMR-DES 和 TMBS-DES 算法的期望时间内调度任务优先级随之增加。这是由于随着任务有效完成时间延长，任务成像机会增加，任务合成机会增加，促使任务期望时间内调度任务优先级增加。此外，RBHA、BS-DES 算法期望时间内调度任务优先级随着 baseDueDate 增大而减小，因为 RBHA、BS-DES 算法不考虑任务合成，有效完成时间的延长增加了任务在其期望完成时间外调度的可能性，所以期望时间内调度任务优先级随之减小。

随着 baseDueDate 增加，扰动测度先是呈现上升趋势。这是由于有效完成时间延长增加了任务的成像机会，接受任务数量增加，使得调度过程中需要更多的任务移位和任务取消操作，因此，扰动测度增加。但是随着任务有效完成时间的进一步延长，任务间冲突减少，使得更多新任务不需要向后移位和任务取消就能被系统接受，所以扰动测度下降，如图 12(c)所示。





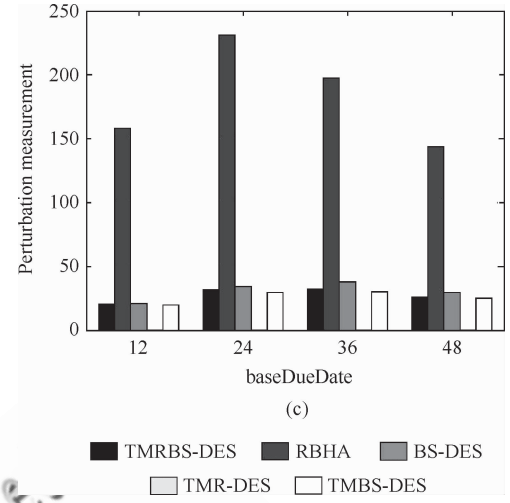


图 12 有效完成时间对算法性能的影响

Fig. 12 Algorithm performance impact of due date

5 结 论

提出了综合考虑任务合成、修复和向后移位的多星应急任务动态调度算法 TMRBS-DES。通过大量模拟实验对 TMRBS-DES 算法的性能进行了测试。测试结果表明 TMRBS-DES 算法能够通过任务合成与修复增加任务收益、减小调度扰动，提高卫星成像效率。此外，TMRBS-DES 算法通过启发式任务插入规则和向后移位策略，增加了调度灵活性，在兼顾调度稳定性的基础上，获得了较好的调度收益。

下一步研究工作主要包括 3 个方面：首先，进一步完善多星应急任务动态调度模型，增加卫星能量、存储容量等约束；其次，将算法进行拓展，解决多星实时调度问题；最后，提出多星动态容错调度算法，有效应对资源失效等问题。

参 考 文 献

[1] Bianchessi N, Cordeau J F, Desrosiers J, et al. A heuristic for the multi-satellite, multi-orbit and multi-user management of earth observation satellites. European Journal of Operational Research, 2007, 177(2): 750-762.

[2] Cordeau J, Laporte G. Maximizing the value of an earth observation satellite orbit. Journal of Operational Research Society, 2005, 56(8): 962-968.

[3] Lin W C, Liao D, Liu C, et al. Daily imaging scheduling of an earth observation satellite. IEEE Transactions on System, Man, Cybernetics, Part A: Systems and Hu-

mans, 2005, 35(2): 213-223.

[4] Zhu K J, Li J F, Baoyin H X. Satellite scheduling considering maximum observation coverage time and minimum orbital transfer fuel cost. Acta Astronautica, 2010, 66(1): 220-229.

[5] Dai G X, Da Q L. The study of combinatorial scheduling problem in emergency systems. Systems Engineering and Electronics, 2000, 20(9): 52-55. (in Chinese)
戴更新, 达庆利. 多资源组合应急调度问题的研究. 系统工程与电子技术, 2000, 20(9): 52-55.

[6] He J M, Liu C L, Cao J, et al. Emergency management and emergency systems—placement, scheduling and algorithms. Beijing: Science Press, 2005. (in Chinese)
何建敏, 刘春林, 曹杰, 等. 应急管理与应急系统——选址、调度与算法. 北京: 科学出版社, 2005.

[7] Bensana E, Verfaillie G, Agnese J C, et al. Exact and inexact methods for the daily management of an earth observation satellite. Proceedings of International Symposium on Space Mission Operations and Ground Data Systems, 1996, 4: 507-514.

[8] Hall N G, Magazine M J. Maximizing the value of a space mission. European Journal of Operational Research, 1994, 78(2): 224-241.

[9] Lin W C, Chang S C. Hybrid algorithms for satellite imaging scheduling. IEEE International Conference on Systems, Man and Cybernetics, 2005, 3: 2518-2523.

[10] Lin W C, Liao D Y. A tabu search algorithm for satellite imaging scheduling. Proceedings of 2004 IEEE International Conference on Systems, Man and Cybernetics, 2004, 2: 1601-1606.

[11] Globus A, Crawford J, Lohn J, et al. Scheduling earth observing fleets using evolutionary algorithms: problem description and approach. Proceedings of 3rd International NASA Workshop on Planning and Scheduling for Space, 2002.

[12] Globus A, Crawford J, Lohn J, et al. A comparison of techniques for scheduling fleets of earth-observing. Journal of Operational Research Society, 2003, 56(8): 962-968.

[13] Pemberton J C, Greenwald L G. On the need for dynamic scheduling of imaging satellites. Proceedings of American Society for Photographing and Remote Sensing (ASPRS-02), 2002.

[14] Liao D Y, Yang Y T. Imaging order scheduling of an earth observation satellite. IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, 2007, 37(5): 794-802.

[15] Liao D Y, Yang Y T. Satellite imaging order scheduling with stochastic weather condition forecast. Proceedings of 2005 IEEE International Conference on Systems, Man and Cybernetics, 2005, 3: 2524-2529.

[16] Billups S C. Final report of the UCDHSC mathematics clinic: satellite mission scheduling with dynamic tasking. Denver: Mathematics Department, University of Colorado at Denver and Health Sciences Center, 2006.

[17] Wang J M, Li J F, Tan Y J. Study on heuristic algorithm for dynamic scheduling problem of earth observation satellites. Proceedings of 8th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2007, 1: 9-14.

[18] Wu G H, Ma M H, Wang H L, et al. Multi-satellite observation scheduling based on task clustering. Acta Aeronautica et Astronautica Sinica, 2011, 32(7): 1275-1282. (in Chinese)
伍国华, 马满好, 王慧林, 等. 基于任务聚类的多星观测调度方法. 航空学报, 2011, 32(7): 1275-1282.

[19] Bai B C, Ci Y Z, Chen Y W. Dynamic task merging in multi-satellites observing scheduling. Journal of System Simulation, 2009, 21(9): 2646-2649. (in Chinese)
白保存, 慈元卓, 陈英武. 基于动态任务合成的多星观测调度方法. 系统仿真学报, 2009, 21(9): 2646-2649.

[20] Bai B C, He R J, Li J F, et al. Imaging satellite observation scheduling with task merging. Acta Aeronautica et Astronautica Sinica, 2009, 30(11): 2165-2171. (in Chinese)
白保存, 贺仁杰, 李菊芳, 等. 考虑任务合成的成像卫星调度问题. 航空学报, 2009, 30(11): 2165-2171.

[21] He R J. Research on imaging reconnaissance satellite scheduling problem. Changsha: National University of Defense Technology, 2004. (in Chinese)
贺仁杰. 成像侦察卫星调度问题研究. 长沙: 国防科学技术大学, 2004.

作者简介:

王建江 男, 博士研究生。主要研究方向: 多目标组合优化, 卫星资源管理与调度。

Tel: 0731-84574552

E-mail: jianjiangwang@nudt.edu.cn

朱晓敏 男, 博士, 讲师。主要研究方向: 云计算, 容错计算, 绿色计算和效能评估。

Tel: 0731-84574552

E-mail: xmzhu@nudt.edu.cn

Multi-satellite Dynamic Scheduling Method for Emergencies

WANG Jianjiang[1], ZHU Xiaomin[1,*], WU Chaobo[1,2], QIU Dishan[1]

1. *Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China*
2. *No.95246 Unit, People's Liberation Army of China, Beihai 536000, China*

Abstract: To solve multi-satellite dynamic scheduling problems in emergency, a multi-objective mathematic programming model is established in this paper. A novel task merging strategy is proposed for multiple imaging satellites: a multi-satellite multi-orbit task merging graph (MSMOTMG) model is established and a task merging algorithm—CP-TM is proposed. In addition, a rehabilitation technique based on task decomposition is suggested to overcome the disadvantage that task merging may cause tasks to have less imaging opportunities. To further enhance the schedulability, the backward shift of tasks in the waiting sequences is considered in our study. Furthermore, a novel dynamic algorithm called TMRBS-DES is presented, which comprehensively considers the task merging, rehabilitation and backward shift. Extensive experiments by simulations are conducted to compare TMRBS-DES with an existing algorithm—RBHA as well as three baseline algorithms—BS-DES, TMR-DES and TMBS-DES. Experimental results demonstrate that TMRBS-DES improves the scheduling quality and is suitable for multi-satellite dynamic scheduling in emergency.

Key words: imaging satellite; dynamic emergency scheduling; mathematical model; task merging; backward shift; rehabilitation; heuristic algorithm

Received: 2012-06-13; **Revised**: 2012-09-14; **Accepted**: 2012-11-29; **Published online**: 2013-01-09 11:15
URL: www.cnki.net/kcms/detail/11.1929.V.20130109.1115.007.html
Foundation items: National Natural Science Foundation of China (61104180, 71271216); National Basic Research Program of China (6136101)

* **Corresponding author**. Tel.: 0731-84574552 E-mail: xmzhu@nudt.edu.cn