

航天器调度中的资源约束推理方法*

廉振宇^{1,3}, 王云², 陈成³, 谭跃进³

(1. 北京系统工程研究所, 北京 100101; 2. 后勤学院, 北京 100101;

3. 国防科技大学 信息系统与管理学院, 湖南 长沙 410073)

摘要:资源水平边界的估计是构建资源利用可行计划中的一个基础问题。通过分析航天器资源约束的共享与分离并存、累积与瞬时消耗并存、过度订阅与区间调度并存等特点,提出了资源时间网络、时间约束网络和约束网络相结合的资源约束描述方法;构建了增量式基于包络的资源约束算法和最早开始时间链展开资源约束算法,以快速获取资源一致的柔性解。实例证明,该方法较好地解决了航天器调度的资源约束推理问题。

关键词:航天器;资源推理;资源时间网络;约束网络;最大流算法

中图分类号:TP 181 **文献标志码:**A **文章编号:**1001-2486(2014)06-106-06

Resource constraint reasoning methods for spacecraft scheduling

LIAN Zhenyu^{1,3}, WANG Yun², CHEN Cheng³, TAN Yuejin³

(1. Beijing Institute of System Engineering, Beijing 100101 China; 2. College of Logistics, Beijing 100101, China;

3. College of Information System and Management, National University of Defense Technology, Changsha 410073, China)

Abstract: The estimation of resource level's envelope is a basic problem in constructing a feasible schedule for resource utilization. A new resource reasoning description method that combining resource temporal network, constraint network and temporal network was proposed by analyzing the resource constraint characteristics, such as coexistence of sharing and detaching resource, coexistence of cumulative and instantaneous consumption, coexistence of over-subscription and interval scheduling resource. Considering the specific characteristic of resource temporal network, a resource model was constructed, and two algorithms of the EBA (Envelop Based Algorithm) based on maximal flow and the ESTA^C (Early Start Time Algorithm with Chaining) based on early start time solution to solve with a resource feasible solution were given. The incremental EBA algorithm and ESTA^C algorithm were presented to get a feasible solution quickly. Finally, some experiments were presented to prove the validity of the methods. Results show that the efficiency and validity of the proposed methods are better than EBA algorithm and ESTA^C algorithm in solving the problems of resource constraint reasoning.

Key words: spacecraft; resource reasoning; resource temporal network; constraint network; maximal flow algorithm

从20世纪60年代末,人们就开始考虑将航天器应用于侦察、跟踪和监视^[1]。资源调度是有效利用航天器完成用户需求的关键,而对资源建模与处理的好坏直接影响到资源调度的优劣。在对传统航天器资源的管控过程中,将航天器观测过程对电量和存储的消耗数量简化建模为持续时间的常数比例^[2]。随着航天器能力的提升,尤其是机动能力和分辨率的提高,航天器对地观测过程对于星上电源和存储等资源的消耗跟当前的系统参数相关。如果按照传统方式处理会对系统效能的提升造成瓶颈,很多航天器任务的调度中具

有这些特性。资源缺陷和冲突的计算及消解是构建资源满足计划的一个基础问题^[3]。

1 航天器调度的资源约束特点

航天器调度中的资源约束具有如下特点:

1) 资源共享与分离并存

航天器资源中,某些资源可以同时被多个活动消耗或者补充,如存储资源、电池资源等。存储资源可以同时用于成像的写入和数传的读取,即实时数传。此时,存储的占用量由两个活动共同决定;同时,某些资源同一时刻只能被一个活动占用,如

* 收稿日期:2013-02-27

基金项目:国家自然科学基金资助项目(71031007,70801062,71071156);湖南省研究生科研创新资助项目(CX2011B023);国防科技大学研究生创新资助项目(B110504)

作者简介:廉振宇(1985—),男,河北永清人,工程师,博士,E-mail:zylian@nudt.edu.cn

相机资源、天线资源等。相机资源同一时刻只能对一个目标成像,即只能被一个成像活动占用。该特征使得两种资源无法单纯利用针对 JobShop、RCPSP 等的经典调度理论处理资源约束。

2)资源在时间上的累积和瞬时消耗并存

执行航天任务过程中,某些活动对资源的消耗在某一时刻瞬间完成,如相机开机和相机关机对相机状态的改变等;某些活动对资源的消耗在某一个持续时间内完成,如数传对航天器天线的占用等;某些活动对资源的消耗是一个累积过程,如数传和成像对航天器存储器的消耗和补充;某些资源在没有活动时以一定速率补充,活动对资源的消耗和补充具有叠加效果,如成像和对日定向对航天器电量的消耗和补充。上述四种情况分别如图 1 中 4 个子图所示。本文在对第三种情况进行处理时,将线性持续消耗集中到最早开始时间点瞬时消耗,即转换为阶梯常量消耗问题^[4]。第四种情况,在本文中暂不考虑。

3)资源过度订阅与区间调度

航天器属于稀缺资源,只能部分满足用户需求。不同于传统调度中任务确定求 Makespan 最小,航天器调度的目标函数是为满足用户需求的收益最大。因此,需要在搜索中选择最优的任务组合,以达到综合效益最优。航天器调度中活动具有较大的时间窗口,具有明显的区间调度特性^[5]。而在搜索中,需要不断增删任务,增删任务过程中,若同时确定活动的开始时间,则会由于近视考虑问题造成过度承诺^[6-7]。因此,既需要在调整中保持时间的柔性,又需要在局部调整时增量式计算资源约束的一致性。

2 航天器调度问题的资源模型

2.1 资源时间网络

Allen^[8]归纳出任意两个区间之间的 13 种关系,进而 Dechter 等^[9]提出时间约束网络,解决了区间关系的标准化求解和一致性问题。在此基础上,Laborie^[10]提出资源时间网络的形式化描述,集成了规划(逻辑的绝对变化)与调度(资源的相对变化和时间)。资源时间网络利用时间约束网络的区间表示特性使其可以处理连续时间上的资源约束,即对资源约束的推理,在鲁棒性调度^[6]、不确定调度^[11]和基于约束的调度^[12]中应用较多。

定义 1 资源:一个资源是一个随时间变化的数值流。该流在时刻 t 的值称为资源在时刻 t 的水平,以 $l(t)$ 表示。

按照资源分类,资源分为:库存资源、占用性资源和时间性资源。占用性资源可以看作是库存资源的特例,即活动的开始消耗资源,活动的结束生产相应的资源。单能力资源可以看作是占用性资源的特例,即资源的容量为 1。

定义 2 资源状态: t_s 和 t_e 表示时间点变量, q 表示常数,则一个资源的状态定义为:

① 资源状态的绝对变化 $A(q, t_s)$,表示资源由当前水平 l 在时间点 t_s 变为 q ;

② 资源状态的相对变化 $R(q, t_s)$,表示资源由当前水平 l 在时间点 t_s 变为 $l + q, q > 0$ 对应于补充 $q, q < 0$ 对应于消耗 q ;

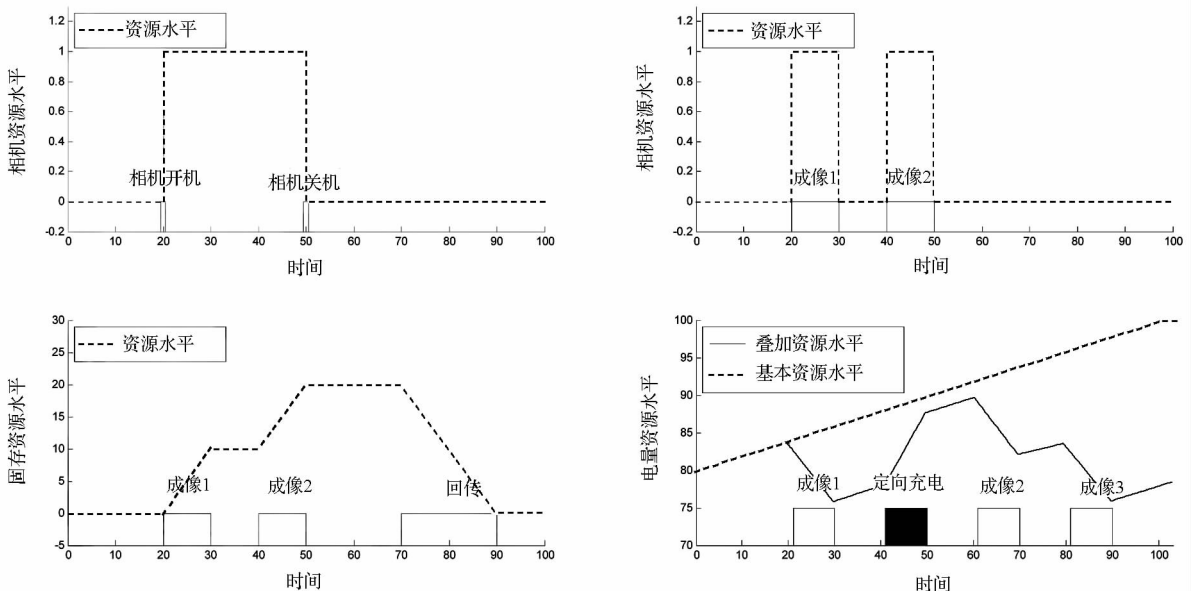


图 1 资源消耗特性示意图

Fig. 1 Resource consumption characteristics

③ 资源状态的不大于条件 $L(q, t_s, t_e)$, 表示资源在时间区间 $[t_s, t_e)$ 保持不大于水平 q ;

④ 资源状态的不小于条件 $G(q, t_s, t_e)$, 资源在时间区间 $[t_s, t_e)$ 保持不小于水平 q 。

A, R, L, G 4 个状态描述了资源的绝对变化、相对变化、不大于条件和不小于条件。若事件 $x \in A \cup R$ 描述资源变化, 则 $t(x)$ 描述其发生时间, $q(x)$ 描述其资源变化量。若 $x \in L \cup G$ 描述资源条件, 则 $t_s(x)$ 描述其发生时间, $t_e(x)$ 描述其结束时间, $q(x)$ 描述 $t_s(x)$ 到 $t_e(x)$ 间的资源要求。 $E(q, t_s, t_e)$ 表示相等条件, 即在时间区间 $[t_s, t_e)$ 保持资源水平 q , 可以理解为 $L(q, t_s, t_e)$ 和 $G(q, t_s, t_e)$ 的并。

定义 3 资源时间网络 (Resource Temporal Network, RTN) 表示为 (T, A, R, L, G, N) , T 是时间点变量集合, A, R, L 和 G 分别对应在 T 上的绝对变化、相对变化、小于和大于条件, N 是由简单时间网络描述的时间关系网络。

一个资源时间网络描述了一个给定数值变量 (资源水平) 在时间上的可能演化集合, 资源时间网络可以描述的问题已经不只是资源水平的变化问题, 还可以涵盖资源数值状态变量的变化问题, 即规划中的绝对变化 A 以及条件 L 和 G 。活动对资源消耗变量与时间点之间的依赖关系在 RTN 中没有直接支持, 但可以通过在约束传播框架下的辅助约束来处理。本文主要对其相对变化 R 进行描述和处理, A 主要描述初始状态, L 和 G 主要描述资源的容量限制。资源时间网络示意图如图 2 所示。

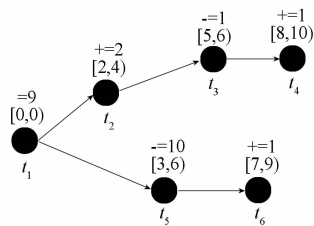


图 2 资源时间网络示意图
Fig. 2 Resource temporal network

定义 4 RTN 的一致性。一个 RTN 满足以下条件, 则认为是一致的:

$$\forall (x_1, x_2) \in A^2, q(x_1) \neq q(x_2) \Rightarrow t(x_1) \neq t(x_2)$$

$$\forall (x_1, x_2) \in A \times R, t(x_1) \neq t(x_2)$$

$\forall \tau \in [t_s(x), t_e(x)]$, 定义 $\bar{\sigma}(\tau)$ 和 $\bar{q}(\tau)$ 分别为时刻 τ 之前最近的一次绝对变化时刻和资源变化数值。则

$$l(\tau) = \bar{q}(\tau) + \sum_{x \in R, \bar{\sigma}(\tau) < t(x) \leq \tau} q(x)$$

$$l(\tau) \leq q(x), \forall x \in L, \forall \tau \in [t_s(x), t_e(x))$$

$$l(\tau) \geq q(x), \forall x \in G, \forall \tau \in [t_s(x), t_e(x))$$

定义 5 资源包络。由一个时间柔性计划在所有可能执行情况下的资源消耗和生产的紧致边界。

资源包络即上述 RTN 一致性条件中的第三条 $l(\tau)$ 。其余两个条件是可以直接检验的, 因此, 主要难点在检验资源包络不超过资源的最大最小能力。计算资源包络可以潜在地节省大量的搜索 (可以早回溯和发现解)。计算资源包络可以平衡计划执行时的资源消耗/生产。

根据 Laborie 的定义^[10], 一个 RTN 的实例与对应的 (Simple Temporal Network, STN) 是一致的, 且其本身是一致的, 则认为当前实例为 RTN 的解。

基于上述描述方式, Laborie 提出了平衡约束 (Balance Constraint, BC) 方法^[13], 算法的复杂性取决于对应 STN 弧一致性维持; Muscettola 提出了基于资源包络 (Envelop Based Algorithm, EBA) 方法^[14], 算法复杂性为 $O(n \times \text{MaxFlow}(n, m) + n^2)$, 其中 n 为事件点数量, m 为在对应的弧一致的 STN 上反序关系数量; Muscettola 又通过在单最大流问题中增加和删除边序, 改进了 EBA 算法^[15] 复杂性为 $O(\text{MaxFlow}(n, m) + n^2)$ 。Nicola 提出了最早开始时间链展开 (Early Start Time Algorithm with Chaining, ESTA^C) 算法^[7], 算法复杂性为 $O(n \times \log(n) + n \times r \times \text{MaxCapacity}_r)$, 其中 r 为资源数量, MaxCapacity 为所有资源最大能力。BC 算法在时间复杂性和效果上没有 EBA 和 ESTA^C 优。EBA 和 ESTA^C 的差别是前者给出的是最坏情况下的资源界, 后者给出的是特定情况下实际的资源界。后者被证明为当前最高效的求界算法^[7,16]。然而该算法针对的是一次资源推理过程, 并没有考虑过度订阅问题的增量调整。

本文主要关注资源上的推理和决策问题, 不考虑如何生成一个资源时间网络, 假设所有的资源状态都是已知的, 一个资源时间网络的决策变量只有时间变量, 其他变量的传播和处理通过约束传播和约束规划的方法处理。

2.2 航天器资源推理问题描述

对于航天器调度问题来说, 多个资源之间的关联主要体现在时间约束和其他约束上, 可以通过时间网络和约束网络的方式处理^[17]。因此, 本文只考虑一个资源的推理子问题。假设针对资源 r , 存在活动集合 Y, T 是时间点变量集合, R 为在 T 上的相对变化, N 是由简单时间网络描述的时

间约束网络,问题为已知 $Y-1$ 的资源时间网络是一致的,验证 Y 的资源时间网络是否存在一致解。我们可以建立约束满足模型如下:

约束条件:

$$q(x) \neq q(y) \Rightarrow t(x) \neq t(y), \quad \forall (x, y) \in A^2 \quad (1)$$

$$t(x) \neq t(y), \quad \forall (x, y) \in A \times R \quad (2)$$

$$t(y) \neq t(x), \quad \forall (y, x) \in A \times R \quad (3)$$

$$0 \leq l(\tau) \leq Q_r, \quad \forall \tau \in [t_s(x), t_e(x)] \quad (4)$$

前提:

$$q(x_1) \neq q(x_2) \Rightarrow t(x_1) \neq t(x_2), \quad \forall (x_1, x_2) \in A^2 \quad (5)$$

$$t(x_1) \neq t(x_2), \quad \forall (x_1, x_2) \in A \times R \quad (6)$$

$$\forall x, x_1, x_2 \in Y-1, y \in Y \cap y \notin Y-1$$

约束条件描述目标状态,前提描述的是 $Y-1$ 的资源状态。其中式(1)~(3)描述新活动对资源的一致性。式(4)描述新活动添加后整体的资源容量不超过限制。式(5)、式(6)描述的是 $Y-1$ 的资源一致性状态。可以更直观地表示为图 3 所示的表示形式。

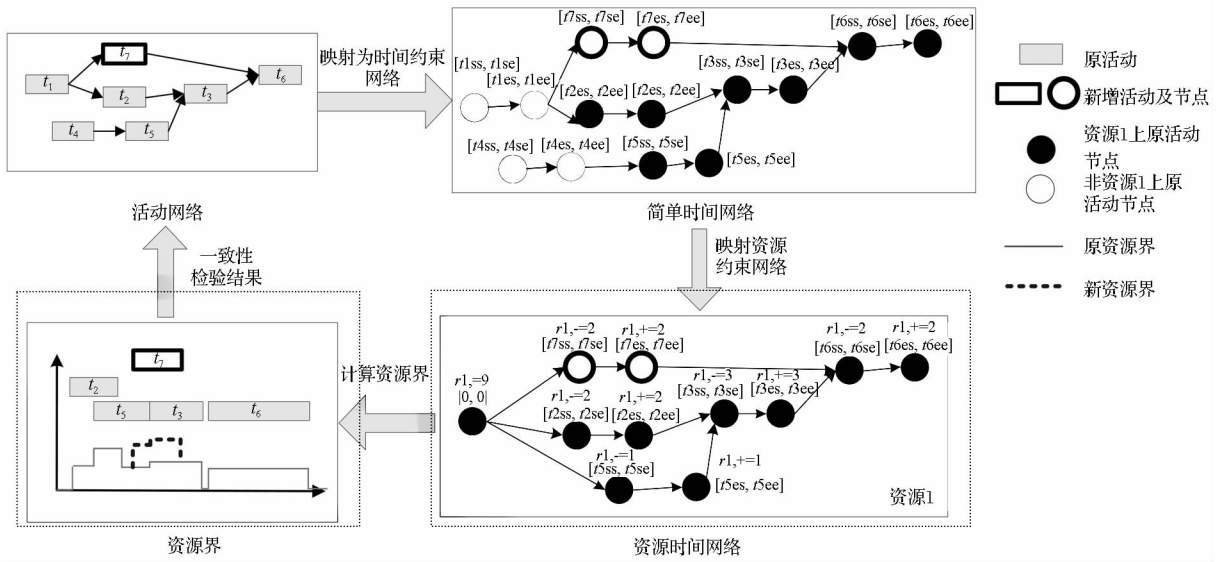


图 3 航天器资源约束推理过程示意

Fig. 3 Process of spacecraft's resource constraint reasoning

3 航天器调度中的资源约束推理方法

以图 3 为例,已知包含活动集合 $\{t_1, t_2, t_3, t_4, t_5, t_6\}$ 的活动网络的时间和资源是一致的,即该活动网络是一致的。现添加活动 t_7 ,需要确定该活动网络的一致性,活动网络的时间一致性可以通过最短路算法检验^[18],活动网络的资源一致性检验是本文研究的重点。

3.1 基于最大流算法的一致性检验

为计算资源包络,确定任一时刻资源水平的最大值,将时刻点分为 3 部分:

B_t : 不在 t 之后的时间点集合, s. t. $lst(t_i) \leq t$;

E_t : t 相关的时间点集合, s. t. $est(t_i) \leq t < lst(t_i)$;

A_t : t 之后的时间点集合, s. t. $est(t_i) > t$ 。

在考虑时刻 t 时,只需要考虑 B_t 和 E_t ,无须考虑 A_t 。 B_t 集合中的事件必然对资源 r_j 分布产生作用。 E_t 集合中的事件有可能产生作用,只有

被安排在 t 之前的活动才产生作用。该活动子集中的活动具有特性: $\forall t_i \in P, t_i < t_i' \Rightarrow t_i' \in P$ 表示若 t_i 属于 P ,且存在活动 t_i 在活动 t_i' 之前的序关系,则 t_i' 也属于 P 。文献[15]证明了通过计算最大流网络 $P_{max} \subseteq E_t$,可以得到在时刻 t 产生影响的最大资源水平变化。事实上,并不需要计算所有的时刻 t ,只需计算那些 B_t 或者 E_t 发生改变的時刻即可。对于任一时间点而言,只能发生在两个时刻点,最早时间 $est(t_i) = -d(t_i, t_0)$ 和最晚时间 $lst(t_i) = d(t_0, t_i)$ 。

然而,在调度算法中使用资源包络方法的主要弊端是最大流算法的计算复杂性问题^[15]。算法描述如图 4 所示。

3.2 基于 ESTA^C算法的一致性检验

ESTA^C算法资源分布为最早开始时间分布,即

$$Q_j^{est}(t) = \sum_{t_i \in T \wedge est(t_i) \leq t} ru_{ij}$$

ru_{ij} 是活动 t_i 对资源 r_j 的改变量。该方法通

```

EBA(P)//输入问题P
POS=TimeConsistCheck(P)//时间一致性检查
Sort(T)//对资源时间网络中时间点集合升序排列
for all timepoint  $e_i$  in  $T$ 
    Divide( $T, e_i, B_e, E_e$ )//按照 $e_i$ 区分 $T$ 中活动
    FG = FlowGraph(POS)//映射为资源增量流图
     $P_{max}$  = MaxFlow(FG)//找到最大流网络
     $P'_{max}$  =MaxFlow(-FG)//反向
     $Lb = \Delta(B_e)+\Delta(P_{max})$ //最小先序集的最大增量
     $Lb' = \Delta(B_e)+\Delta(P'_{max})$ //下界
    if ( $Ub > \text{MaxCapacity}(r_j)$ )
        ||  $Lb < \text{MinCapacity}(r_j)$ )
            return Null
Return POS

```

图 4 EBA 算法伪代码

Fig. 4 Pseudocode of EBA algorithm

过一个精确时间解(最早开始时间解)得到一个资源分布。该方法与 EBA 方法不同,EBA 针对的是所有时间行为可能的范围,ESTA^C针对的是某个特定场景。算法描述如图 5 所示。

```

ESTAC(P)//输入问题P
S=ESTAC(P)//根据最早开始时间算法求的一个解
POS=P//偏序解初始化
Sort(T)//对解中活动集合升序排列
for all activity  $t_i$  in  $S$ 
    for all resource  $r_j$ 
         $k=1$ 
        for 1 to  $ru_{ij}$ //活动 $i$ 占用资源 $j$ 的量
             $t = \text{LastElement}(\text{queue}_{jk})$ 
            while  $s_{it} \geq e_i$ //找到不在 $t_i$ 之前活动
                 $k=k+1$ 
                 $t = \text{LastElement}(\text{queue}_{jk});$ 
            POS=POS  $\cup \{t_i\}$ 
            Enqueue( $\text{queue}_{jk}, t_i$ )
             $k=k+1$ 
TimeConsistCheck(POS)
Return POS

```

图 5 ESTA^C算法伪代码

Fig. 5 Pseudocode of ESTA^C algorithm

该算法中使用一系列的队列 $queue_{jk}$ 表示资源 r_j 的每一个能力单元。首先将解 S 中的活动按照开始时间排序。接着为每一个活动分配需要的能力单元,仅选择在活动开始时间可用的能力单元。当一个活动分配到一个队列,则在当前活动与队列中的前一个活动之间添加一个新的约束。

3.3 增量式一致性检验方法

对于资源时间网络,EBA 算法的一致性检验算法^[15]考虑的是整个资源时间网络重新计算资源界的问题,复杂性为 $O(\text{MaxFlow}(n, m) + n^2)$ 。当前最大流算法最坏时间复杂度^[19]为 $O(n \ln n)$,算法效率的提高可以直接反映到一致性检验上。ESTA^C算法效率比 EBA 有所提升,仍是基于约束调度中的瓶颈。但在实际应用中,由于航天器调

度是过度订阅的,活动网络需要迭代调整,这种局部调整问题,尚没有形成针对该问题有效的求解算法。因此,本文只考虑增量式地计算资源界,提出增量式最大流算法。

增量式的主要思想是根据活动增减、时间变量值域的变化和资源消耗数量的变化等通知,动态更新资源时间网络,记录更新节点。同时,利用累积消耗资源的向后传播效应,实现下一次检验则从更新节点向后检验,避免了更新节点前的节点重复检验。对于瞬时消耗或者占用资源,则只需要检查当前活动的最早开始时间与最晚完成时间之间的一致性即可。

4 实例验证

本文在实验中采用的数据是按照航天器 IKONOS 的轨道和姿态性能,由 STK (Satellite Tool Kit) 软件仿真得到的。本文提出的航天器调度中的资源约束推理是一个约束满足问题模型,目标是找到满足资源约束的一致性时间赋值。因此对于不同方法而言,求解时间是算法优劣的重要指标,具体解的优劣在本文不作区分。计算结果如图 6 所示,给出了本文提出的增量式一致性检验方法 IEBA 和 IESTA^C与原一致性检验方法在节点数量不断增长的情况下计算时间的比较,可以得到本文方法在节点数目少的情况下,两种算法差别不大,但随着节点数目的不断增多,本文方法在计算时间上明显优于原 EBA 和 ESTA^C算法。

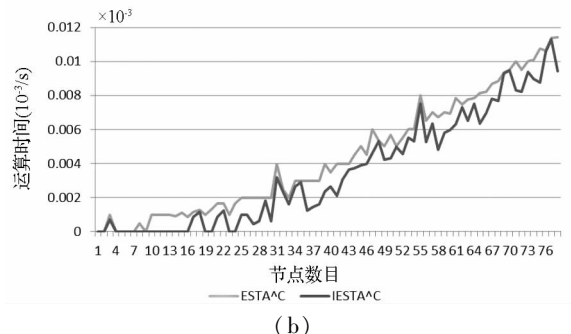
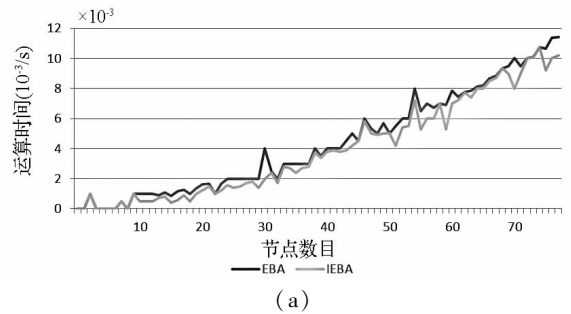


图 6 算法性能比较分析

Fig. 6 Comparison of algorithm performance

5 结论

本文总结了航天器调度中资源约束的特点:资源共享与分离并存,使得问题具有级联效应,不能局部解决;资源在时间上的累积和瞬时消耗并存,使得问题具有后效性。资源过度订阅与区间调度,使得问题具有增量求解的必要性。因此,本文提出在 EBA 和 ESTA^C算法基础上的增量式一致性检查算法。通过仿真实验,证明本文提出的方法是可行且有效的。同时,本文方法基于资源时间网络描述实现,并不局限于敏捷卫星调度领域,只要是能够描述为本文资源时间网络描述的资源约束推理问题都可以采用本文方法进行求解。

本文主要针对的是常量资源消耗的情况,未来在资源约束推理方面还可以考虑阶梯常量、线性和非线性消耗的情况。

参考文献 (References)

- [1] 贺仁杰,李菊芳,姚峰,等. 成像卫星任务规划技术[M]. 北京:科学出版社,2010.
HE Renjie, LI Jufang, YAO Feng, et al. Mission planning techniques for earth observation satellites[M]. Beijing: Science Press, 2010. (in Chinese)
- [2] 贺仁杰. 成像侦察卫星调度问题研究[D]. 长沙:国防科学技术大学,2004.
HE Renjie. Research on earth observation satellite scheduling problem [D]. Changsha: National University of Defense Technology,2004. (in Chinese)
- [3] Pollicella N, Smith S. F, Cesta A, et al. Steps toward computing flexible schedules [C]. Online Workshop at CP, Kinsale, Ireland, 2003.
- [4] Frank J. Bounding the resource availability of partially ordered events with constant resource impact [C]. LNCS 3258. Berlin Heidelberg, Springer-Verlag, 2004:242 - 259.
- [5] Laborie P, Rogerie J, Shaw P, et al. Interval-based language for modeling scheduling problems: an extension to constraint programming [M]. In Algebraic Modeling Systems, Applied Optimization, J. Kallrath, Springer-Verlag: Berlin Heidelberg, 2012, 6:111 - 143.
- [6] Policella N, Cesta A, Oddi A, et al. Solve and robustify: synthesizing partial order schedules by chaining[J]. Journal of Scheduling, 2009, 12: 299 - 314.
- [7] Policella N, Cesta A, Oddi A, et al. From precedence constraint posting to partial order schedules[J]. AI Communications, 2007, 20 (3): 163 - 180.
- [8] Allen J F. Maintaining knowledge about temporal intervals[J]. Communications of the ACM, 1983, 26 (11): 832 - 843.
- [9] Dechter R, Meiri I, Pearl J. Temporal constraint networks[J]. Artificial Intelligence, 1991, 49: 61 - 94.
- [10] Laborie P. Resource temporal networks: definition and complexity [C]//Proceedings of the 18th International Joint Conference on Artificial Intelligence, 2003:948 - 953.
- [11] Fu N, Lau H C, Varakantham P, et al. Robust local search for solving RCPSP/max with durational uncertainty[J]. Journal of Artificial Intelligence Research, 2012, 43 (1): 43 - 86.
- [12] Oddi A, Cesta A, Policella N, et al. Iterative flattening search for resource constrained scheduling[J]. J. Intelligent Manufacturing, 2010, 21 (1): 17 - 30.
- [13] Laborie P. Algorithms for propagating resource constraints in AI planning and scheduling: existing approaches and new results[J]. Artificial Intelligence, 2003, 143 (1): 151 - 188.
- [14] Muscettola N. Computing the envelope for stepwise-constant resource allocations [C]. The 8th International Conference on the Principles and Practices of Constraint Programming, 2002:139 - 154.
- [15] Muscettola N. Incremental maximum flows for fast envelope computation [C]. International Conference on Automated Planning and Scheduling (ICAPS - 04). Whistler, British Columbia, Canada, 2004.
- [16] Banerjee D, Haslum P. Partial-order support-link scheduling [C]. The 21st International Conference on Automated Planning and Scheduling. Freiburg, Germany, 2011:307 - 310.
- [17] Lian Z Y, Tan Y J, Chen Y W, et al. A constraint-based framework for AEOS Mission planning and scheduling [C]// Proceedings of ICAPS - 12 Workshop on Constraint Satisfaction for Planning and Scheduling. Sao Paul, Brazil, 2012.
- [18] 廉振宇,谭跃进,严珍珍. 敏捷卫星调度的时间约束推理方法[J]. 系统工程与电子技术, 2013, 6.
LIAN Zhenyu, TAN Yuejin, YAN Zhenzhen. Temporal constraint reasoning methods for AEOS scheduling [J]. System Engineering and Electronic Technology, 2003, 6. (in Chinese)
- [19] Hochbaum D. The pseudo-flow algorithm and the pseudo-flowbased simplex for the maximum flow problem [C]. IPCO - 98. Houston, Texas, 1998:325 - 337.