

基于分层约束图的多节点探测器任务规划方法

付康, 赵清杰, 杨和星

(北京理工大学计算机学院, 北京 100081)

摘要: 针对多节点探测器的附着任务规划问题, 提出了一种基于分层约束图的约束可满足任务规划方法。介绍了时间资源约束的表示方式、任务时间网络和分层约束图模型, 并将该问题转化为约束可满足问题。该方法通过任务时间网络推理变量的值域信息, 并采用基于约束双向支持的弧相容算法进行约束传播。设计了结合层级约束信息的变量启发式规则以及优先满足资源约束的值启发式规则。实验结果验证了所提出方法的有效性。

关键词: 多节点探测器; 任务规划; 分层约束图; 约束可满足问题; 启发式规则

中图分类号: TP18

文献标识码: A

文章编号: 2096-9287(2025)03-0305-10

DOI: 10.3724/j.issn.2096-9287.2025.20240063

引用格式: 付康, 赵清杰, 杨和星. 基于分层约束图的多节点探测器任务规划方法[J]. 深空探测学报(中英文), 2025, 12(3): 305-314.

Reference format: FU K, ZHAO Q J, YANG H X. Multi-node probe task-planning method based on hierarchical constraint graph[J]. Journal of Deep Space Exploration, 2025, 12(3): 305-314.

引言

开展小天体探测研究是理解太阳系的形成演化、了解生命起源和物质发展过程, 以及发现处理潜在危险小天体等的有效途径^[1]。现有小天体探测活动中大多采用单节点构型探测器, 针对小天体表面引力弱造成探测器不能稳定附着的问题, 中国学者提出了柔性连接的多节点构型探测器, 并对小天体附着任务进行了研究^[2-4]。多节点构型探测器中的每个节点均具有一定的感知能力和机动能力, 各个节点之间相对独立但又存在约束限制。相较于单节点构型的探测器, 多节点探测器具有更强的感知能力和协同能力, 方便附着于具有微小重力的小天体表面。

深空探测器附着任务指的是深空探测器通过自身携带的科学载荷如成像仪、光谱仪等对途中或周边的小天体进行一定程度的探索, 并尝试将探测器附着于小天体表面。如何对任务序列进行合理规划是实现探测器稳定附着的基础。深空环境具有未知、约束复杂、时间、资源约束耦合紧密等特性。在进行任务规划时, 探测器不仅要考虑任务的选择, 还要对时间、资源进行合理分配, 以保证探测任务顺利完成^[5]。目前的深空探测任务规划方法主要包括: 受自然群体行为启发并具有分布式协同特征的群智能优化方法^[6-8]、依靠与环

境交互进行自主决策的强化学习方法^[9-10]、聚焦于处理复杂任务约束的基于模型转换的任务规划方法^[11-13]以及紧密结合任务时间属性特点的时态规划方法^[14-17]。

相对于单节点探测器的任务规划问题, 多节点探测器的任务规划问题涉及多个层次的时间约束: 节点间约束、子系统间约束和任务间约束。多节点探测器任务规划问题的难点在于不同层次之间的复杂约束耦合, 极大增加了规划的难度。因此需要研究更加高效的方法, 以应对多节点探测器任务规划中的挑战。本文针对多节点探测器的附着任务规划问题, 提出了一种基于分层约束图的约束可满足任务规划方法。利用约束可满足算法处理约束的优势, 将该任务规划问题转换为约束可满足问题。根据约束层级信息设计了相关的启发式策略, 并结合约束推理传播技术, 以快速获取满足多节点探测器时间和资源约束的规划序列方案。仿真实验表明, 该算法能够有效减少约束检验次数, 并提高求解多节点探测器任务规划序列的效率。

1 问题描述与建模

1.1 问题描述

多节点探测器附着任务规划问题是指如何合理安排任务序列, 以满足多节点探测器内部的复杂时间和资源约束。如图1所示, 探测器的各节点包含不同的子

系统, 不同的子系统中可执行的任务单元集合也是不一致的, 且约束存在于各任务单元之间。该问题形式化描述为: $\phi = \langle O, A, C, I, G \rangle$ 。其中 O 描述了多节点探测器各节点内部子系统集合, 包含规划过程中需要考虑的所有子系统; A 为探测器各子系统可执行的最小任务单位集合; C 为探测器约束集合, 主要包括时间约束和资源约束; I 为附着任务规划问题的初始状态; G 为附着任务规划问题的目标状态。多节点探测器附着任务规划需要在满足约束集合 C 的前提下, 从初始状态 I 出发, 选择恰当的探测器各节点、各子系统的执行方案, 从而实现目标状态 G 。

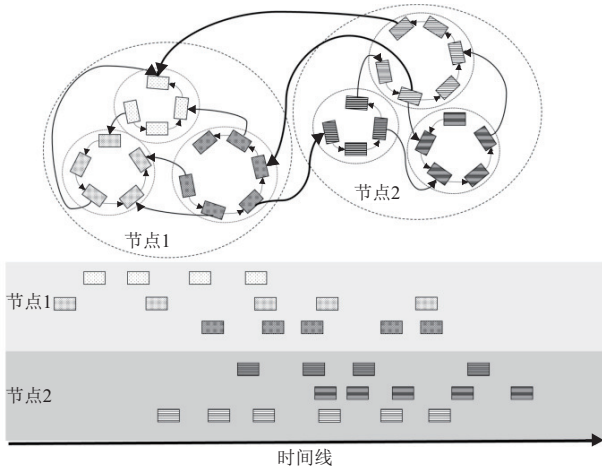


图1 多节点探测器附着任务规划问题示意图
Fig. 1 Diagram of task planning for multi-node probe landing

1.2 问题建模

1.2.1 时间约束与分层约束图

多节点探测器由多个组件构成, 一个活动通常需要多个组件的任务共同配合完成, 因此任务之间存在时间上的约束。每个任务都可以通过两个时间对象来描述, 对于任务 $task = \langle t_1, t_2 \rangle$, t_1 、 t_2 分别表示任务开始时间对象和结束时间对象。时间对象 $t = \{node_id, system_id, task_id, time\}$, 其中 $node_id$ 、 $system_id$ 、 $task_id$ 和 $time$ 是时间对象 t 的属性, 分别表示时间对象 t 所属的节点、子系统、任务编号以及所对应的时间。因此, 任务之间的时间约束可以通过时间对象之间的约束来表示。例如对于任务1 $task_1 = \langle start_1, end_1 \rangle$, 任务2 $task_2 = \langle start_2, end_2 \rangle$, 如果任务2执行前需要先执行任务1, 则时间约束 $c_{task} = \langle start_2, end_1 \rangle = [down, up]$ 表示任务2要在任务1完成后 $[down, up]$ 时间区间内执行, 如式 (1) 所示。

$$down \leq start_2 - end_1 \leq up \quad (1)$$

依据不同时间对象之间的关联性, 即时间对象所归属的任务、子系统和节点的不同特性, 将约束边进行细粒度的划分, 并构建了一个适用于多节点探测器系统的分层约束图模型 $C = \{C_{task}, C_{system}, C_{node}\}$, 如图2所示。其中, C_{task} 表示同一子系统内任务之间的相互约束, C_{system} 描述不同子系统之间的约束关系, 而 C_{node} 则代表节点层面的约束情况。在图2中, 3种箭头分别代表3类不同的时间约束, 分别是节点间约束、子系统间约束和任务间约束。

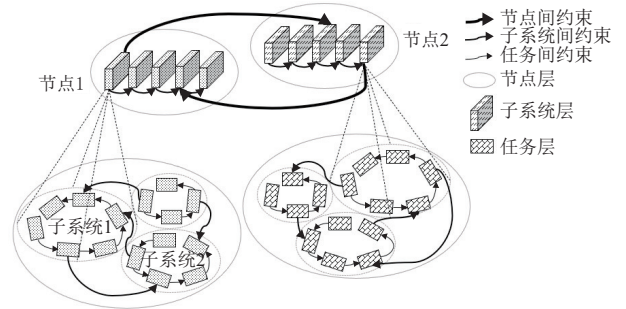


图2 分层约束图示意图
Fig. 2 Diagram of the hierarchical constraint graph

1.2.2 任务时间网络

在上述时间约束的表示方式中, 任务之间的约束通过时间对象之间的关联加以描述。同一任务的开始时间对象和结束时间对象之间存在潜在的约束关系, 即任务的结束时间与开始时间之间的时间差需满足特定范围内的要求。因此, 本质上多节点探测器图模型中的约束可以进一步简化。在图论中, 有向图存在入度的概念, 指的是以该顶点为终点的有向边数量。在此基础上, 构建如图3所示的任务时间网络。

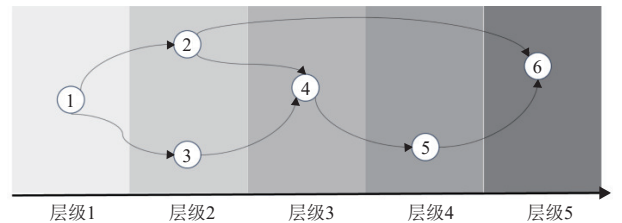


图3 任务时间网络示意图
Fig. 3 Diagram of the task time network

层级计算方法如下。

- 1) 入度为0的时间对象顶点的层级数为1;
 - 2) 入度为1的时间对象顶点的层级数为其约束有向边的另一端时间对象顶点的层级数加1;
 - 3) 入度大于1的时间对象顶点的层级数为其约束有向边的另一端时间对象顶点集合中最大层级数加1。
- 例如图3中, 顶点1的入度为0, 其层级数为1; 顶

点2、3的入度均为1,其层级数为顶点1的层级数加1,即为2;而对于顶点6而言,其入度为2,有向边另一端的顶点包括4和5,顶点4的层级数为3,顶点5的层级数为4,顶点6的层级数为有向边另一端的顶点集合最大层级数4加1,即5。

1.2.3 资源约束

多节点探测器在任务执行过程中涉及两类关键资源:独占资源和持续消耗资源。独占资源同一时刻仅允许一个节点使用,形成节点间的互斥约束,例如通信轨道资源,如式(2)所示。持续消耗资源则包括电量、带宽和散热资源等。这些资源可以同时供多个节点使用,但在执行任务时会按固定速率消耗。不同任务的资源消耗速率可能不同,但所有节点在同一时刻的资源消耗总和必须保持在规定的上限之内,如式(3)所示。

$$\sum_{i=1}^N R_{\text{type1}}^j(i,t) \leq 1, \quad j = 1, 2, \dots, N_{\text{type1}} \quad (2)$$

$$\sum_{i=1}^N R_{\text{type2}}^j(i,t) \leq U_j, \quad j = 1, 2, \dots, N_{\text{type2}} \quad (3)$$

其中: N 表示多节点探测器的节点数目; N_{type1} 、 N_{type2} 分别代表独占资源和持续消耗资源的数目; $R_{\text{type1}}^j(i,t)$ 表示 t 时刻第 i 个节点对第 j 种独占资源的占有情况,取值为1代表正在占用该资源; $R_{\text{type2}}^j(i,t)$ 表示 t 时刻第 i 个节点对第 j 种持续消耗资源的消耗速率; U_j 代表第 j 种持续消耗资源消耗速率的上限值。

1.2.4 多节点探测器附着任务规划模型

从减少资源消耗和提高执行效率的角度出发,本研究以最小化多节点探测器附着任务的完成时间为优化目标,如式(4)所示。该任务规划问题的约束条件如式(5)所示,涵盖独占资源约束、持续消耗资源约束以及时间约束。

$$\min f = \max(t_i) \quad (4)$$

$$\begin{cases} \sum_{i=1}^N R_{\text{type1}}^j(i,t) \leq 1, & j = 1, 2, \dots, N_{\text{type1}} \\ \sum_{i=1}^N R_{\text{type2}}^j(i,t) \leq U_j, & j = 1, 2, \dots, N_{\text{type2}} \\ \sum_C c_{ij} = 0 \end{cases} \quad (5)$$

其中: t_i 为任务规划问题中时间对象的时间值; C 为多节点探测器系统图模型; c_{ij} 为时间约束损失代价值,如果 t_i 、 t_j 满足约束边 e_{ij} , c_{ij} 取值为0,否则取值为1。

以上描述了多节点探测器附着任务规划模型,以最短任务完成时间为优化目标,在满足所有时间约束、持续消耗资源和独占资源约束的前提下,探究多节点探测器内部任务序列的合理构建方案。

2 约束可满足基本理论

约束可满足技术可以自然地处理多个约束,且具有较强的适应性,可以处理线性约束、逻辑约束等不同的约束类型。引入约束可满足的思想,凭借其过滤、消解约束的优势,处理多节点探测器附着任务规划问题。

2.1 问题定义

约束可满足问题(Constraint Satisfaction Problem, CSP)主要由以下3个部分构成:

- 1) 变量集 $X = \{x_1, x_2, \dots, x_i, \dots, x_w\}$,其中 w 为变量数目;
- 2) 值域集 $Y = \{y_1, y_2, \dots, y_i, \dots, y_w\}$,其中 y_i 为 x_i 的值域集合;
- 3) 约束集 $C = \{c_1, c_2, \dots, c_i, \dots, c_u\}$,其中 u 为约束的数量。

根据约束所描述的限制关系中涉及的变量数目将约束划分为不同类型:限制变量数目为1的约束为一元约束,限制变量数目为2的约束为二元约束;限制任意变量数目的约束为全局约束。而任意有限值域的约束都可以通过引入足够的约束变量的方式转化为二元约束,所以可以将任何约束可满足问题转化为只含二元约束的约束可满足问题。

当存在一组变量的取值能够满足约束集中所有约束时,称其为约束可满足问题的一个解,而当不存在这样的变量取值组合时,该问题是无解的。约束可满足规划的目标就是找到一组值,能够给每一个变量赋值,且该组值满足约束集中的所有约束条件。

2.2 约束传播算法

约束传播算法是求解约束可满足问题最常见的有效手段之一,核心思想是利用约束间的相互影响即局部相容性实现搜索空间的削减。常见的相容类型包括结点相容、弧相容、路径相容和全局约束。结点相容是指单个变量的值域满足该变量的所有一元约束,此时称该变量是结点相容的。在约束可满足问题中,一个二元约束可以由两条弧形式化描述。例如 x_i 和 x_j 之间存在一个二元约束,该二元约束可以由弧 $\langle x_i, x_j \rangle$ 和 $\langle x_j, x_i \rangle$ 来表示。如果 x_i 确定被赋予值域 y_i 中的某个值 val_i ,而 x_j 能够在值域 y_j 中找到满足二元约束的取值 val_j ,即 x_i 对弧 $\langle x_i, x_j \rangle$ 相容。路径相容涉及到多个二元

约束之间的潜在约束, 如果 x_i 对弧 (x_i, x_{mid}) 相容, x_{mid} 对弧 (x_{mid}, x_j) 相容, 那么 x_i, x_{mid}, x_j 这条路径是相容的。全局约束是一种独特的约束, 往往通过专用算法去处理。而根据上述提到的相容类型特点, 构建发展了一系列的约束传播算法: 结点相容算法、弧相容算法和路径相容算法。其中弧相容算法凭借其简单有效, 剪枝能力强等优点, 被广泛应用。

弧相容算法是求解约束可满足问题过程中的常用算法, 用以检验约束可满足问题是否存在解。如果存在约束可满足变量不满足弧相容, 那么该约束可满足问题无解, 仅当约束可满足变量均满足弧相容时, 该问题才可能存在解。而如果此时约束可满足变量均被赋值, 那么该组赋值组合即为问题的一个可行解。搜索过程中往往会将弧相容算法作为推理的核心内容, 用于指导搜索过程, 减小搜索空间从而提高搜索效率。

在实际应用中, 使用最多的弧相容算法为AC-3^[18]和AC-4^[19]弧相容算法。虽然AC-4弧相容算法的最优时间复杂度优于AC-3弧相容算法, 但是AC-4弧相容算法的实际平均时间复杂度接近最差时间复杂度, 且具有较高的空间复杂度。另外, AC-4弧相容算法对于变量值域较大且约束较弱的问题的处理效果没有AC-3好。因此, 本文采用AC-3弧相容算法作为任务规划方法推理部分的核心内容。

2.3 求解约束可满足问题的回溯搜索算法

回溯搜索算法是求解约束可满足问题的一种完备的核心算法, 很多约束可满足算法都是在该算法的基础上进行启发式、弧相容算法的设计从而实现问题求解效率的提升。部分赋值的回溯算法以深度优先搜索的形式为基础架构, 不断为变量赋值, 直到出现不满足弧相容的情况, 进行回溯操作。部分赋值的回溯搜索算法的具体步骤如下:

步骤1: 判断当前变量是否均已被赋值。如果是, 则搜索到约束可满足问题的解, 返回真; 否则, 执行步骤2。

步骤2: 选择未赋值的变量, 将该变量标记为已赋值。

步骤3: 对选中的变量的值域内的值进行排序。

步骤4: 判断选中的变量是否还有未尝试的值。如果是, 执行步骤5; 否则, 将该变量标记为未赋值, 返回假。

步骤5: 为选中的变量赋值, 进行约束满足检验, 如果检验结果为真, 执行步骤6; 否则跳转到步骤4。

步骤6: 进行约束传播推理操作。如果推理结果为真, 返回真; 否则, 将该赋值标记为已尝试并跳转到

步骤4。

其中, 常见的优化改进出现在步骤2、步骤3和步骤6。步骤2可以设计有效的变量启发式, 尽可能早地选择更加重要的变量; 步骤3中可以设计合理的值启发式, 为上一步选中的变量赋予更合适的值; 步骤6则是可以引入对约束传播推理技术的各种改进策略。另外, 还有一部分相关研究的优化方向为改进回溯算法的回跳机制, 当弧相容不满足的情况出现时, 尽快回跳到消解冲突的决定性变量, 从而实现搜索效率的进一步提高。

本文在设计多节点探测器附着任务规划问题的求解算法时也主要考虑了以上优化策略, 设计了合理的变量启发式和值启发式, 采用高效的弧相容算法, 结合基础的回溯搜索算法, 构建了完整的算法架构。

3 基于分层约束图的约束可满足任务规划方法

本文提出的基于分层约束图的约束可满足任务规划方法, 采用了基于任务时间网络的值域推理来初始化约束可满足变量的值域范围, 结合约束可满足方法的优化方向, 分别设计了结合约束双向支持性的AC-3弧相容算法、基于分层约束的变量启发式和优先考虑资源约束满足的值启发式, 以实现任务规划序列的快速求解。

3.1 基于任务时间网络的值域推理

在多节点探测器附着任务规划问题中, 需要求解各节点、各子系统内的执行序列, 即子系统执行最小任务单元的时间区间分布情况。在建模的过程中, 通过任务开始时间对象和任务结束时间对象来描述该任务执行时间的区间范围。理论上, 开始时间对象和结束时间对象的时间取值范围为 $[0, \infty]$ 。而约束可满足问题的变量往往是值域有限的, 且变量值域的大小会在一定程度上影响问题的求解速度。一般来说, 在其它条件相同的情况下, 约束可满足变量的值域越大, 问题求解难度越高。因此, 在任务时间网络模型的基础上, 以开始时间对象为约束可满足变量, 设计相关值域推理规则, 以获取变量的初始值域范围。由于仅将开始时间对象定义为约束可满足变量, 时间对象之间的约束需要统一转化为开始时间对象间的约束关系并构成约束集的一部分。

在进行约束可满足变量的值域推理之前需要进行约束转换操作, 结合时间约束的描述与定义和任务开始时间对象与任务结束时间对象间的潜在约束关系等信息, 设计了约束转换规则。例如, 任务 $task_1$ 的持

续时间为 d_1 , 任务 $task_2$ 的持续时间为 d_2 , 对由时间对象描述的任务 $task_1 = \{task_{1s}, task_{1e}\}$, $task_2 = \{task_{2s}, task_{2e}\}$ 而言, 存在4种类型的约束形式:

- 类型1: 开始时间对象与开始时间对象之间的约束;
- 类型2: 开始时间对象与结束时间对象之间的约束;
- 类型3: 结束时间对象与开始时间对象之间的约束;
- 类型4: 结束时间对象与结束时间对象之间的约束。

假定时间对象间的约束限制为在前的时间对象减去后面的时间对象的差值范围为 $[down, up]$ 。这4种约束类型的转换规则分别为:

类型1: 符合预期约束形式, 无需执行转换操作;

类型2: 不符合预期约束形式, 需要通过 $task_{2s}$ 表示 $task_{2e}$, 将约束 $down \leq task_{1s} - task_{2e} \leq up$ 转换为 $down + d_2 \leq task_{1s} - task_{2s} \leq up + d_2$;

类型3: 不符合预期约束形式, 需要通过 $task_{1s}$ 表示 $task_{1e}$, 将约束 $down \leq task_{1e} - task_{2s} \leq up$ 转换为 $down - d_1 \leq task_{1s} - task_{2s} \leq up - d_1$;

类型4: 不符合预期约束形式, 需要通过 $task_{1s}$, $task_{2s}$ 表示 $task_{1e}, task_{2e}$, 将约束 $down \leq task_{1e} - task_{2e} \leq up$ 转换为 $down - d_1 + d_2 \leq task_{1s} - task_{2s} \leq up - d_1 + d_2$ 。

将所有约束按照以上规则统一转化为预期约束形式后, 利用任务时间网络, 对时间对象即约束可满足变量的值域进行推理。值域推理规则为:

- 1) 构建源点作为时间基准, 为源点与层级数为1的时间对象构建约束, 作为推理的初始化信息;
- 2) 按照任务时间网络中时间对象的层级数从低到高进行值域的推理;
- 3) 对于入度为1的时间对象, 通过有向约束边另一侧的时间对象值域范围和约束信息来推理该时间对象的值域范围;
- 4) 对于入度大于1的时间对象, 根据与之相关联的每一条约束边另一端的时间对象的值域范围和约束边信息推理对应的值域限制范围, 形成多个对应约束关系的值域范围集合, 对值域范围集合进行交集的求解, 如果存在交集, 该交集即为时间对象的值域范围, 否则取值域范围集合中最小值和最大值分别作为推理所得的时间对象值域的下限值和上限值。

通过基于任务时间网络的值域推理算法, 初始化作为约束可满足变量的时间对象的值域范围, 利用约束之间的相互影响, 有效地削减了搜索空间, 为后续进一步求解提供了良好的基础。

3.2 结合约束双向支持性的弧相容算法

约束可满足变量值域更新时, 约束传播算法能够削减其它相关约束可满足变量的值域范围, 减小搜索

空间, 从而提高求解问题的速度。另外, 由于约束传播算法的核心是局部相容性, 确保了所获得的解的一致性和正确性, 即解是满足约束可满足问题中的所有约束的。合理高效的约束传播算法能够提升约束可满足算法的性能。本节从既有约束传播算法和约束性质等角度出发, 采用了结合约束双向支持性的AC-3弧相容算法。

凭借简单、高效、适用性高等优点, AC-3弧相容算法被广泛应用于各种类型的约束可满足问题。AC-3弧相容算法会将约束可满足问题中所有约束构成的弧放入队列; 逐一取出弧, 判断约束可满足变量对于该弧是否相容, 如果相容就继续取下一条弧, 否则删除该变量值域中不相容的值, 并将与该变量存在约束关系的约束可满足变量的相关弧均放入队列中; 如果执行删除操作后变量的值域为空, 则该约束可满足问题无解^[18]。在AC-3弧相容算法执行过程中, 对于弧 $\langle x_i, x_j \rangle$ 进行检验操作, 若 x_i 对弧 $\langle x_i, x_j \rangle$ 不相容, 将弧 $\langle x_j, x_i \rangle$ 放入队列中, 而当 x_j 对弧 $\langle x_j, x_i \rangle$ 也不相容时, 执行删值操作使之相容, 并将弧 $\langle x_i, x_j \rangle$ 放入队列中, 而这时 x_i 对弧 $\langle x_i, x_j \rangle$ 实际上已经相容。因此, AC-3弧相容算法在这种情况下会多执行一次检测操作。

考虑到上述AC-3弧相容算法处理过程中的这一现象, 引入约束双向支持性^[20]以提高约束传播算法效率。约束双向支持性指的是若 x_i 取值为 $value_i$, x_j 取值为 $value_j$ 时满足 x_i 与 x_j 之间的约束, 那么 x_i 取值为 $value_i$ 是弧 $\langle x_i, x_j \rangle$ 中 x_j 取值为 $value_j$ 的一个支持, 且 x_j 取值为 $value_j$ 是弧 $\langle x_j, x_i \rangle$ 中 x_i 取值为 $value_i$ 的一个支持。在AC-3弧相容算法处理约束的过程中双向考虑弧的相容性, 顺序检测处理弧 $\langle x_i, x_j \rangle$ 和 $\langle x_j, x_i \rangle$, 且在相容情况下不再将弧 $\langle x_i, x_j \rangle$ 加入队列从而实现检测次数的减少。

综上所述, 采用结合约束双向支持性的AC-3弧相容算法, 顺序处理约束相关的弧检测操作, 减少了约束传播推理过程中的检验次数, 提高了推理检验算法的效率。

3.3 启发式设计

合理利用有效信息的启发式规则能够提高搜索算法的求解效率。在2.3节提及了约束可满足问题的回溯搜索结构, 搜索过程与推理过程交替进行以加速可行解的获取。而约束可满足问题的回溯搜索方法仍存在一些待优化的步骤, 包括如何选取将被赋值的变量、如何为选中的变量进行赋值操作等。

3.3.1 变量启发式设计

在回溯搜索过程中需要从未赋值的约束可满足变量集合中选择某个变量, 并为该变量执行赋值等后续

操作。对于包含 n 个变量的约束可满足问题而言,选取变量的顺序方案数为 $n!$ 。因此,如何安排变量的选取顺序加快搜索过程成为当前优化的一个重要方向。而变量启发式的作用就是利用问题特定的领域知识确定选取变量的策略,为当前搜索选择变量,尽快找到问题的解。

约束可满足问题的通用信息包括变量自身值域范围,变量之间的约束关系等。最小剩余值启发式和最大度启发式等就是利用了这些通用信息的经典启发式。最小剩余值启发式从变量值域大小出发,优先选择变量值域最小的变量,其估值函数为

$$\text{score}(x_i) = |Y_i| \quad (6)$$

但是最小剩余值启发式在该问题搜索过程中的指导效果并不理想。这是由于该约束可满足问题中约束比较稀疏,拥有较多约束边的变量对问题的求解起着关键性的作用。而剩余值较小的变量并不能够直接地削减相关变量的值域范围,且不相容情况也会在搜索过大部分变量后才出现,不能够较为直观地体现约束间的相互影响。最大度启发式从变量关联程度的角度考虑,选取与其它变量间存在约束数目最多的变量,其估值函数为

$$\text{score}(x_i) = \sum c_i \quad (7)$$

其中, $\sum c_i$ 表示与约束可满足变量 x_i 相关的约束边数目。

在任务规划问题的建模过程中,考虑多节点探测器分级协作机制,构建了分层约束图,将约束划分为不同层级。不同层级的约束的重要程度是不一样的,比如节点间约束对变量值的选择影响范围比子系统间约束更大。因此,考虑将分层约束信息融合到启发式规则中,设计了基于分层约束的变量启发式规则,估值函数为

$$\text{score}(x_i) = w_n \times \sum c_{ni} + w_s \times \sum c_{si} + w_t \times \sum c_{ti} \quad (8)$$

其中: w_n 、 w_s 、 w_t 分别为节点间约束、子系统间约束和子系统内任务间约束的权重系数,取值依次为3、2、1; $\sum c_{ni}$ 、 $\sum c_{si}$ 、 $\sum c_{ti}$ 分别表示与变量 x_i 相关的节点间约束、子系统间约束和子系统内任务间约束数目。

3.3.2 值启发式设计

当确定了需要赋值的约束可满足变量后,如何从该变量的值域中挑选最优值进行后续搜索成为优化的重点。处理和排序变量的值域有助于削减其它变量的

值域空间,从而加快求解速度。此外,在处理变量值域时,还需考虑资源约束,以确保最终解满足所有资源限制。对值域的处理和排序可以设计为启发式规则,这类专注于值选择的启发式称为值启发式。

在将任务规划问题转化为约束可满足问题的过程中,资源约束作为全局约束引入,并在值启发式中进行处理。设定问题的资源跟踪量,用以跟踪资源在搜索过程中的变化情况,并由此推理判断被选择变量满足资源约束的可行取值范围。资源跟踪量随时间变化的示意图如图4所示。其中, P_{trace} 、 Q_{trace} 和 B_{trace} 为资源跟踪量,记录回溯搜索过程中已赋值变量消耗资源的量或速率的总和; P_{max} 、 Q_{max} 和 B_{max} 表示资源量的最大值或消耗速率的最大值,因此资源最大值减去资源跟踪量的值为当前的资源剩余值。利用资源跟踪量削减变量值域的策略为:在取值为 value_i 时,只有当所有资源的剩余值均大于或等于变量所对应任务的消耗资源的量或速率时,该值 value_i 才会保留,否则从变量值域中删去该值。

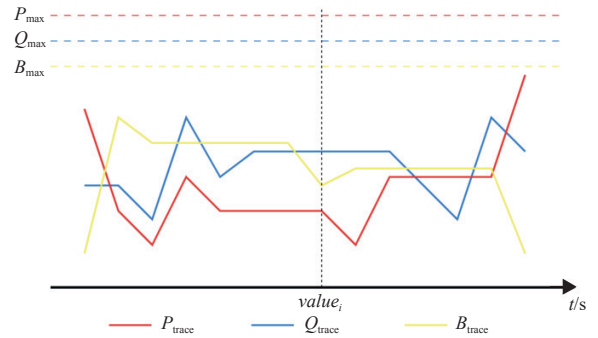


图4 资源跟踪量时间变化示意图

Fig. 4 Diagram of resource tracking volume changes over time

从最短任务完成时间的角度出发,在上述考虑资源约束的值域处理策略的基础上,采用最小预期值策略,即对变量值域内的可行值从小到大进行排序,构建了优先考虑资源约束满足的值启发式规则。

3.4 算法步骤

针对多节点探测器的附着任务规划问题,基于分层约束图的任务规划方法引入了约束可满足的思想。该方法利用任务时间网络推理变量的值域范围,并将有效的启发式规则与约束传播算法融合到回溯搜索的主体结构中,从而提升了算法的求解效率。该算法的具体步骤如下:

步骤1: 初始化资源跟踪量和编码为约束可满足变量的时间对象在任务时间网络中的层级信息;

步骤2: 采用基于任务时间网络的值域推理方法,推理变量的值域范围,并记录变量赋值情况;

步骤3: 根据基于分层约束的变量启发式选择未赋值变量。如果所有变量均已被赋值, 返回真, 记录的变量取值即为约束可满足问题的解; 否则执行步骤4;

步骤4: 依据优先考虑资源约束满足的值启发式从值域中选择值赋值给变量, 如果被选中变量的值域中所有值均已尝试过, 标记该变量未被赋值, 返回假; 否则执行步骤5;

步骤5: 判断变量取值是否满足与已赋值变量间的约束, 如果不满足, 跳转到步骤4, 否则, 标记该变量已被赋值, 更新资源跟踪量并执行步骤6;

步骤6: 采用结合约束双向支持性的弧相容算法进行约束传播推理, 如果推理结果为真, 跳转到步骤3, 否则变量值域和资源跟踪量状态回退并跳转到步骤4。

4 实验与分析

为了验证本文提出的多节点探测器任务规划方法, 设计了10个具有不同规模的多节点探测器系统的仿真测试算例, 依次编号为1~10。其中各个算例所涉及的时间约束数目分别为: 59、86、95、125、152、179、212、248、269、302。每个算例输入主要由5个文件组成, 分别是系统规模文件、任务执行时间文件、独占资源文件、持续消耗资源文件和时间约束文件。5个文件所表示的内容如下:

- 1) 系统规模文件: 说明算例的节点、子系统和任务的数量;
- 2) 任务执行时间文件: 说明算例中每个任务的执行时间;
- 3) 独占资源文件: 说明算例中使用独占资源的任务列表;
- 4) 持续消耗资源文件: 说明算例中使用持续消耗资源的任务列表以及消耗速率;
- 5) 时间约束文件: 说明算例中任务之间存在的约束关系。

为了准确评价基于分层约束图的任务规划方法, 实验过程中将其重复运行10次, 并以10次运行的平均运行时间和约束检测次数作为算法的实验结果。

在最大度经典启发式的基础上, 本文考虑了多节点探测器分级特点, 构建了分层约束图, 将约束的分层信息融合到启发式规则之中。为了进一步验证基于分层约束的变量启发式, 进行了变量启发式的对比实验。基于分层约束的变量启发式和最大度启发式的算法运行时间对比结果如表1所示, 检测约束数目对比结果如表2所示。从表1中可以看出随着多节点探测器系统规模的增长, 算法运行时间大体上呈现上升趋势。

对于大部分仿真测试算例而言, 采用了基于分层约束的变量启发式的算法的运行时间是要短于采用最大度启发式的算法的, 这说明了基于分层约束的变量启发式对算法求解速率的提升是高于最大度启发式的, 验证了所提出的变量启发式的有效性。

表1 采用不同变量启发式的算法运行时间对比

Table 1 Comparison of algorithm running times using different variable heuristics

算例编号	基于分层约束的变量启发式	最大度启发式	ms
1	15	15	15
2	31	31	31
3	15	82	82
4	81	81	81
5	62	70	70
6	178	315	315
7	415	596	596
8	535	675	675
9	765	921	921
10	1583	1632	1632
平均值	368	442	442

注: 黑体数值表示实验最优结果。

表2 采用不同变量启发式的算法约束检测次数对比

Table 2 Comparison of algorithm constraint detection times using different variable heuristics

算例编号	基于分层约束的变量启发式	最大度启发式
1	7 418	7 256
2	46 535	48 040
3	31 850	32 270
4	112 222	112 222
5	58 568	59 048
6	123 612	125 383
7	204 262	205 745
8	360 140	358 866
9	490 531	465 112
10	913 369	843 004
平均值	234 851	225 695

注: 黑体数值表示实验最优结果。

在表2中, 随着多节点探测器系统规模的增长, 算法检测约束的次数快速增长。当系统规模较小时, 采用了基于分层约束的变量启发式的算法的约束检测次数是少于采用最大度启发式的算法的; 而当系统规模变大时, 前者要大于后者, 且基于分层约束的变量启发式算法的平均约束检测次数高于最大度启发式算法。这可能是由于基于分层约束的变量启发式优先解决的是关键约束所对应的弧的相容性, 而前期约束传播过程中弧相容算法对变量值域的削减能力较弱, 因此导致检测约束的次数较多, 但是当解决了关键约束的相容性问题后, 问题的求解速度得到大幅度提升, 因此展现出基于分层约束的变量启发式运行时间较短的结果。

本文提出的基于分层约束图的约束可满足任务规划方法,具体的改进模块包括:基于任务时间网络的值域推理,结合约束双向支持性的AC-3弧相容算法,基于分层约束的变量启发式和优先考虑资源约束满足的值启发式。而由于不添加结合约束双向支持性的AC-3弧相容算法模块时,无法在可接受时间内求解该问题,已体现了该模块的有效性。具体的消融实验算法和对应的模块如表3所示,基础的算法结构采用了基于分层约束的变量启发式的回溯搜索,在表中记为回溯搜索。

表3 消融实验对比算法

Table 3 Ablation experiment comparison algorithm				
算法名称	回溯搜索	弧相容算法	值域推理	值启发式
算法a	√	√		
算法b	√	√	√	
算法c	√	√		√
算法d	√	√	√	√

在表4所展示的算法运行时间实验结果中,可以看出添加了基于任务时间网络的值域推理和优先考虑资源约束满足的值启发式两个模块的算法d的运行时间是最短的。由算法b、算法c求解时间和算法a的求解时间差值大小可以推出基于任务时间网络的值域推理模块在提高算法求解速度上的作用是更大的。尤其是对于编号为2的仿真测试算例而言,值域推理模块使得该算例能够在可接受的时间范围内被求解,而仅添加了值启发式模块的算法c则无法做到这一点。这可能是由于在大部分约束可满足问题中,变量的值是不具有特殊信息的,所以值启发式的效果较弱。另外,伴随着多节点探测器系统规模的增长,优先考虑资源约束满足的值启发式所带来的效率的提升并不会有很大起伏,而基于任务时间网络的值域推理模块起到的影响效果则越发明显。

表4 消融实验运行时间对比

Table 4 Comparison of ablation experiment running times				
算例编号	算法a	算法b	算法c	算法d
1	46	15	15	15
2	—	940	—	31
3	122	18	114	15
4	332	109	288	81
5	496	67	487	62
6	1 032	208	720	178
7	1 455	422	1 311	415
8	1 945	744	1 711	535
9	3 716	940	3 488	765
10	4 607	2 190	4 504	1 583

注:—表示无法在可接受时间范围内求解;黑体数值表示实验最优结果。

在表5所展示的算法运行过程中约束的检验次数的对比实验结果中,可以看出优先考虑资源约束满足的值启发式对减少约束检测次数的作用是很小的,而基于任务时间网络的值域推理模块则在减少约束检验次数这一点上体现了显著的优越性,且随着问题规模的增大,其在算法运行过程中减少的约束检验次数就越多,具有很好的扩展性。

表5 消融实验约束检测次数对比

Table 5 Comparison of constraint detection times in ablation experiments

算例编号	算法a	算法b	算法c	算法d
1	25 061	7 418	25 601	7 418
2	—	490 531	—	46 535
3	140 616	31 850	140 616	31 850
4	313 026	112 222	313 026	112 222
5	541 829	58 568	541 829	58 568
6	606 603	123 612	606 603	123 612
7	828 201	204 262	828 201	204 262
8	1 229 170	360 140	1 229 170	360 140
9	2 106 441	490 531	2 106 441	490 531
10	2 563 250	913 369	2 563 250	913 369

注:—表示无法在可接受时间范围内求解;黑体数值表示实验最优结果。

综上所述,基于任务时间网络的值域推理模块和优先考虑资源约束的值启发式模块都能够很好地优化完善算法的性能。基于任务时间网络的值域推理模块在提高算法求解速率和减少算法运行过程中的约束检测次数等方面均具有优秀的表现,而优先考虑资源约束满足的值启发式模块则是能够在一定程度上提高算法的求解速度,确保算法所求得解是满足全局资源约束的,且能够在加入值域推理模块后进一步提升算法的性能。

5 结论

本文针对多节点探测器的附着任务规划问题,提出了一种基于分层约束图的约束可满足任务规划方法。首先对问题进行描述和建模,定义其约束条件,并构建了分层约束图和任务时间网络模型。鉴于多节点探测器任务中复杂的约束条件以及时间和资源约束的紧密耦合,最终将该任务规划问题转化为约束可满足问题。该方法通过提取时间约束信息构建任务时间网络,并推理约束可满足变量的初始值域信息。采用基于双向支持性的弧一致算法作为核心的约束推理传播技术,同时设计了结合约束层级信息的变量启发式规则和优先满足资源约束的值启发式规则,缩小变量的值域范围。仿真结果证明了提出的方法能够快速有效地求解多节点探测器任务规划问题。

参考文献

- [1] 崔平远,陆晓萱,朱圣英,等.小天体柔性附着状态协同估计方法[J].*宇航学报*,2022,43(9):1219-1226.
CUI P Y, LU X X, ZHU S Y, et al. Cooperative state estimation method for small celestial body flexible landing[J]. *Journal of Astronautics*, 2022, 43(9): 1219-1226.
- [2] 崔平远,张成宇,朱圣英,等.小天体柔性附着技术[J].*宇航学报*,2023,44(6):805-816.
CUI P Y, ZHANG C Y, ZHU S Y, et al. Technologies for flexible landing on small celestial bodies[J]. *Journal of Astronautics*, 2023, 44(6): 805-816.
- [3] 崔平远,赵冬越,朱圣英,等.小天体主动附着制导与控制技术研究进展[J].*宇航学报*,2021,42(9):1057-1066.
CUI P Y, ZHAO D Y, ZHU S Y, et al. Research progress of guidance and control technologies for active landing on small celestial bodies[J]. *Journal of Astronautics*, 2021, 42(9): 1057-1066.
- [4] 王棒,徐瑞,李朝玉,等.小天体表面探测器弹跳运动与路径规划[J].*深空探测学报(中英文)*,2022,9(4):447-454.
WANG B, XU R, LI Z Y, et al. Bouncing motion and path planning of small body surface rover[J]. *Journal of Deep Space Exploration*, 2022, 9(4): 447-454.
- [5] 徐瑞,李朝玉,朱圣英,等.深空探测器自主规划技术研究进展[J].*深空探测学报(中英文)*,2021,8(2):111-123,109-110.
XU R, LI Z Y, ZHU S Y, et al. Research progress of autonomous planning technology for deep space probes[J]. *Journal of Deep Space Exploration*, 2021, 8(2): 111-123, 109-110.
- [6] ZHEN Z Y, CHEN Y, WEN L D, et al. An intelligent cooperative mission planning scheme of UAV swarm in uncertain dynamic environment[J]. *Aerospace Science and Technology*, 2020, 100: 105826.
- [7] LU Z D, SHEN X, LI D R, et al. Multiple super-agile satellite collaborative mission planning for area target imaging[J]. *International Journal of Applied Earth Observation and Geoinformation*, 2023, 117: 103211.
- [8] HILTON S, THANGAVEL K, GARDI A, et al. Intelligent mission planning for autonomous distributed satellite systems[J]. *Acta Astronautica*, 2024, 225: 857-869.
- [9] QI Y, GU D, LIU Y, et al. A mission planning method for deep space detectors using deep reinforcement learning[J]. *Aerospace Science and Technology*, 2024, 153: 109417.
- [10] 孙泽翼,王彬,胡馨月,等.深空探测器多智能体强化学习自主任务规划[J].*深空探测学报(中英文)*,2024,11(3):244-255.
SUN Z Y, WANG B, HU X Y, et al. Multi-agent reinforcement learning autonomous task planning for deep space probes[J]. *Journal of Deep Space Exploration*, 2024, 11(3): 244-255.
- [11] 姜啸,徐瑞,朱圣英.基于约束可满足的深空探测任务规划方法研究[J].*深空探测学报(中英文)*,2018,5(3):262-268.
JIANG X, XU R, ZHU S Y. Research on task planning problems for deep space exploration based on constraint satisfaction[J]. *Journal of Deep Space Exploration*, 2018, 5(3): 262-268.
- [12] ZHAO Y T, XU R, JIANG H, et al. Decentralized privacy-preserving onboard mission planning for multi-probe system[J]. *Acta Astronautica*, 2021, 179: 130-145.
- [13] 柳景兴,王彬,毛维杨,等.深空探测器任务规划认知图谱及多属性约束冲突检测[J].*深空探测学报(中英文)*,2023,10(1):88-96.
LIU J X, WANG B, MAO W Y, et al. Cognitive graph for autonomous deep space mission planning and multi-constraints collision detection[J]. *Journal of Deep Space Exploration*, 2023, 10(1): 88-96.
- [14] 李朝玉,徐瑞.一种基于时标状态的启发式航天器任务规划算法[J].*深空探测学报(中英文)*,2015,2(1):20-26.
LI Z Y, XU R. Time stamped states based heuristic algorithm for spacecraft mission planning[J]. *Journal of Deep Space Exploration*, 2015, 2(1): 20-26.
- [15] 金颢,徐瑞,崔平远,等.基于扩展状态深空探测器任务规划方法[J].*深空探测学报(中英文)*,2018,5(6):569-574.
JIN H, XU R, CUI P Y, et al. Mission planning approach based on extensible states for deep space probes[J]. *Journal of Deep Space Exploration*, 2018, 5(6): 569-574.
- [16] 赵凡宇,徐瑞,崔平远.启发式深空探测器任务规划方法[J].*宇航学报*,2015,36(5):496-503.
ZHAO F, XU R, CUI P. Heuristic mission planning approach for deep space explorer[J]. *Journal of Astronautics*, 2015, 36(5): 496-503.
- [17] 赵宇庭,徐瑞,李朝玉,等.基于动态智能体交互图的深空探测器任务规划方法[J].*深空探测学报(中英文)*,2021,8(5):519-527.
ZHAO Y T, XU R, LI Z Y, et al. Mission planning based on dynamic agent interaction graph for deep space probes[J]. *Journal of Deep Space Exploration*, 2021, 8(5): 519-527.
- [18] MACKWORTH A K. Consistency in networks of relations[J]. *Artificial Intelligence*, 1977, 8(1): 99-118.
- [19] MOHR R, HENDERSON T C. Arc and path consistency revisited[J]. *Artificial Intelligence*, 1986, 28(2): 225-233.
- [20] BESSIERE C, FREUDER E C, RÉGIN J C. Using constraint metaknowledge to reduce arc consistency computation[J]. *Artificial Intelligence*, 1999, 107(1): 125-148.
- 作者简介:
付康(2002-),男,博士生,主要研究方向:多智能体协同规划,强化学习。
通信地址:北京市海淀区中关村南大街5号北京理工大学计算机学院(100081)
E-mail: fukang@bit.edu.cn
赵清杰(1966-),女,教授,博士生导师,主要研究方向:机器视觉、机器学习与人工智能、智能体系统等。本文通信作者。
通信地址:北京市海淀区中关村南大街5号北京理工大学计算机学院(100081)
E-mail: zhaobj@bit.edu.cn
杨和星(1999-),女,硕士,主要研究方向:任务规划与重规划。
通信地址:北京市海淀区中关村南大街5号北京理工大学计算机学院(100081)
E-mail: yhx19990702@163.com

Multi-Node Probe Task-Planning Method Based on Hierarchical Constraint Graph

FU Kang, ZHAO Qingjie, YANG Hexing

(School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China)

Abstract: In this paper, a constraint satisfaction-based task planning method based on the hierarchical constraint graph was proposed for the attachment task planning of a multi-node probe. First, the representation of time-resource constraints, task-time networks, and hierarchical constraint graph models were introduced, and the planning problem was transformed into a constraint satisfaction problem. The method inferred the value range information of variables through the task-time network and employed an arc consistency algorithm with bidirectional constraint support for constraint propagation. In addition, the variable heuristic rules guided by constraint hierarchy information and value heuristic rules that prioritize resource satisfaction were designed. Experimental results demonstrate the effectiveness of the proposed method.

Keywords: multi-node probe; task planning; hierarchical constraint graph; constraint satisfaction problem; heuristic rule

Highlights:

- The attachment task planning of a multi-node probe is reformulated as a constraint satisfaction problem by modeling time-resource constraints and a hierarchical constraint graph.
- A constraint satisfaction-based task planning method is proposed based on the hierarchical constraint graph.
- The proposed method is tested by numerous experiments using multiple example data, and the results prove that the task planning method based on hierarchical constraint graph is effective.

[责任编辑: 宋宏, 英文审校: 宋利辉]