

基于动态智能体交互图的深空探测器任务规划方法

赵宇庭^{1,2}, 徐 瑞^{1,2}, 李朝玉^{1,2}, 朱圣英^{1,2}

(1. 北京理工大学 宇航学院, 北京 100081; 2. 深空自主导航与控制工信部重点实验室, 北京 100081)

摘 要: 面对日益复杂的深空探测任务和动态多变的深空环境, 深空探测器需要更高效的任务规划技术以快速生成规划方案。探测器内部各子系统具有分布并行的特征, 可抽象为多智能体系统进行规划。而现有的多智能体规划方法无法直接应用于需要处理时间资源等数值约束的深空探测器任务规划中。针对上述问题, 提出基于分布式求精搜索的多智能体规划空间规划方法, 设计动态智能体交互图引导多智能体协同规划, 将时间资源约束处理抽象为约束满足问题并采用基于图论的方法进行处理。实验表明, 对于包含多子系统的深空探测器的任务规划问题, 所提方法能够有效提高规划效率。

关键词: 深空探测器; 多智能体规划空间规划; 数值约束

中图分类号: TP18

文献标识码: A

文章编号: 2096-9287(2021)05-0519-09

DOI:10.15982/j.issn.2096-9287.2021.20210020

引用格式: 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.

Reference format: 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.

引 言

随着空间中航天器数量增长和航天任务的日益复杂, 地面站的管控压力不断增大, 从而产生了对航天器规划技术的需求。地面自动规划能快速生成规划方案, 提高航天器的管控效率; 在轨自主规划能赋予航天器自我管理能力和减轻地面管控压力。

国内外机构和学者已经对航天器规划技术展开研究。美国国家航空航天局(National Aeronautics and Space Administration)开发了可扩展标准化远程操作规划框架(Extensible Universal Remote Operations Planning Architecture, EUROPA)^[1-2], 该框架内的时间网络模块负责传播时间约束并保持规划解的时间一致性, 资源管理模块负责处理存储能源等数值资源约束。EUROPA在多个航天器上得到了应用, 包括哈勃望远镜的观测调度^[3], “深空一号”(Deep Space 1)^[4]和“地球观测一号”(Earth Observing-1)的自主控制^[5]。

自主规划和调度环境(Automated Scheduling and Planning Environment, ASPEN)^[6]是美国喷气推进实验室(JPL)的人工智能小组开发的一个可重构规划调

度框架, 采用基于冲突修复的规划方法。连续活动调度规划执行和重规划系统(Continuous Activity Scheduling Planning Execution and Replanning, CASPER)^[7]是ASPEN的软实时星上规划版本, 能够在环境变化或目标变化后快速进行重新规划。

以上3个规划系统都采用状态时间线知识表示方法, 适合描述探测器的时间资源约束知识, 可通过在搜索算法中加入启发式提高规划效率。金颢等^[8]提出了基于状态转移图的启发式加速规划搜索。王晓晖等^[9]设计时间线启发式因子引导规划扩展过程, 提高任务规划效率。

除了设计启发式, 利用深空探测器子系统分布并行的特性, 将子系统抽象为智能体, 采用多智能体任务规划方法, 也能够提高规划效率。在领域无关规划(domain-independent planning)的研究中, 多智能体规划是一个重要分支。Brafman和Domshlak^[10]量化表示了MA-STRIPS(Multi-Agent Stanford Research Institute Problem Solver)问题中智能体之间的耦合程度, 证明多智能体规划问题的复杂度主要与智能体间耦合程度相关。MADLA(Multi-Agent Distributed and Local

Asynchronous)^[11]规划器针对MA-STRIPS问题提出了分布式状态空间前向链式多启发式搜索方法。FMAP(Forward Multi-Agent Planning)^[12]为了表示状态变量采用PDDL3.1描述领域知识,并提出一种启发式前向链式偏序规划方法。这些规划器虽然致力于解决多智能体规划问题,但缺乏对时间、资源等数值约束的表达和处理能力^[13]。徐瑞等^[14]提出了一个用于航天器的多智能体规划器,能够处理时间资源约束,采用有中心的多智能体分布式规划,设置管理智能体作为其它智能体通信和协作的中心。

深空探测器与地球距离遥远,传统的规划后再上传指令的控制方式会造成较长的时间延迟,因此需要研究自主规划技术以提高探测器的决策效率和任务收益。深空探测器任务规划存在领域知识复杂、时间资源约束复杂、子系统多、任务目标多等难点。

本文的研究目的是探索应用于复杂多系统探测器的高效规划方法,主要思想是结合多智能体规划高效率的优势和经典航天器规划器的数值处理能力,提出基于动态智能体交互图的深空探测器任务规划架构与方法。

本文建立基于状态时间线的多智能体规划问题模型,并为实现探测器各系统活动及数值约束知识的表示而设计了相应的多智能体规划建模语言。提出基于分布式求精搜索的多智能体规划空间规划方法,不设置管理智能体作为协调中心,而是设计动态智能体交互图引导多智能体协同规划,并采用基于图论的方法处理探测器的时间和资源约束。最后,通过仿真实验验证了方法的有效性。

1 探测器多智能体规划问题建模

由于状态时间线模型具有复杂时间资源约束的描述能力,较为适合深空探测器规划问题建模,本文在时间线模型基础上建立多智能体规划问题模型。

定义1. 状态变量是表示智能体状态的变量,可定义为 $S = \langle N, V, C, D, R \rangle$ 。

其中: N 为状态变量的名称,每个智能体由一个状态变量表示其状态,因此与智能体的名称一致; V 为状态变量的值域,即所有此状态变量可以选取的状态; C 为状态变量各值之间的约束条件,约束条件包括本智能体内部约束和与其它智能体相关的外部约

束; D 为状态变量各取值的持续时长; R 为状态变量各取值对资源变量的影响。

定义2. 状态是状态变量的取值,可定义为 $s = \langle N_s, t_s, t_e, d, r, C_s \rangle$ 。

其中: N_s 为状态的名称; t_s 、 t_e 、 d 为状态开始时刻、结束时刻和持续时长,都表示为区间形式,即 $t_s = [t_s^{\min}, t_s^{\max}]$ 、 $t_e = [t_e^{\min}, t_e^{\max}]$ 、 $d = [d^{\min}, d^{\max}]$; r 为该状态对资源变量的影响,不影响资源的状态不考虑此参数; C_s 为此状态的约束,包含此状态定义的与其它状态的约束关系,这里的其它状态既包括本状态变量的其它取值,也包括属于其它状态变量的状态。

定义3. 状态时间线是状态变量不同取值在时间上的连续分布,表示智能体的状态在规划周期内随时间变化的情况,可定义为 $T = \{s_0, s_1, s_2, \dots, s_n\}$ 。

定义4. 智能体是一个规划单元,依据问题需求和领域知识进行规划,可定义为 $a_i = \langle N_i, \Pi_i, T_i, M_i \rangle$ 。

其中: N_i 为智能体名称; Π_i 为规划问题分解到该智能体的部分; T_i 为该智能体的状态时间线; M_i 为规划方法。

定义5. 多智能体规划问题: $A = \{a_i\}_{i=1}^n$ 为包含 n 个多智能体的集合, S 为所有智能体的状态变量集合, I 为各状态变量初始值集合, G 为各状态变量目标值集合,多智能体规划问题可表示 $P = \langle S, I, G, D \rangle = \{\Pi_i\}_{i=1}^n$ 。对于每一个智能体 a_i ,规划问题为 $\Pi_i = \langle S_i, I_i, G_i, D \rangle$ 。

其中: S_i 为属于智能体 i 的状态变量; I_i 为属于智能体 i 的状态变量的初始值集合; G_i 为属于智能体 i 的状态变量的目标值集合; D 为规划周期时长,规划器只做出周期内的方案,而不是按规则无限扩展方案。规划解由各智能体的状态时间线表示。

对于深空探测器任务规划,探测器的子系统和载荷可抽象为智能体,探测器各系统的活动间的约束关系通过状态的约束集合 C_s 表示。

在具体的规划实现中,需要设计建模语言,用建模语言描述领域知识和规划问题作为规划器的输入。NDDL(New Domain Description Language)是EUROPA采用的建模语言,具备对逻辑约束、时间约束和资源约束的表达能力,适合状态时间线模型的知识表示。本研究对NDDL进行多智能体化的扩展,设计多智能体数值领域建模语言(Multi-Agent Numeric Domain Modelling Language, MANDML)。

MANDML继承了NDDL的功能，并加入多智能体系统所需的元素。

首先，加入智能体的概念，将探测器的子系统抽象为智能体。在规划器输入中添加智能体定义文件，其中包含所有智能体的名称和编号，作为规划过程中智能体相互通信的标识。假设探测器有载荷系统（LoadSys）和姿态系统（AttitudeSys）两个系统需要规划，那么可以定义两个智能体。智能体定义文件的内容如下所示：

Agent 0: AttitudeSys;
Agent 1: LoadSys;

然后，设计适用于多智能体规划的文件输入模式。规划智能体输入相同的智能体定义文件和模型文件，从而使每个规划智能体对探测器的各系统组成和状态间的约束关系有一致的认识。各规划智能体的问题文件是独立的，只包含当前智能体的初始状态和目标状态。比如，对于Agent 1，输入的只有载荷分系统的初始和目标状态。以包含两个规划智能体的多智能体规划系统为例，两个规划智能体的所有的输入如图1所示。

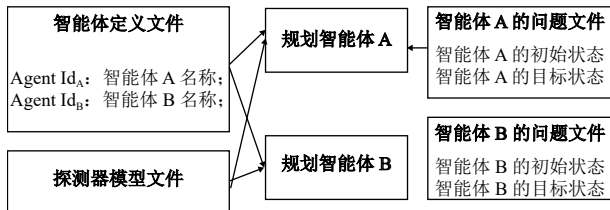


图 1 规划智能体输入文件示意
Fig. 1 Inputs of planning agents

2 多智能体规划空间规划方法

提出一种多智能体规划空间规划方法。不设置管理智能体协调各智能体规划，而是设计动态智能体交互图作为多智能体交互的规则，引导规划搜索的路径在多智能体间扩展。构建多智能体时间约束网络模型，采用图的最短路径算法实现时间约束传播与解耦。构建资源约束网络模型，采用基于图最大流的算法处理资源约束。

2.1 动态智能体交互图

智能体基于状态间的约束关系在规划中展开协作。在当前部分规划中成功加入一个状态后，此状态

的约束集合 C_s 中的其它状态也要随之加入到部分规划当中。例如，相机拍照之前需要姿态校准，那么相机智能体的拍照状态加入规划后，姿态智能体的校准状态也要加入规划，以满足拍照状态与校准状态的约束关系。

定义6.如果状态 a 出现在状态 s 的约束集合 C_s 中，即 $a \in C_s$ ，那么 a 为 s 的**从属状态**， s 为 a 的**主导状态**。

定义7.如果一个状态 a 是状态 s 的从属状态，状态 s 属于智能体 A ，但状态 a 不属于智能体 A ，即 $a \in C_s(s \in V_{S_A}, a \notin V_{S_A})$ ，那么 a 为一个**公共状态**，否则为**私有状态**。

其中： V_{S_A} 表示智能体 A 的状态变量的值域。

在某些多智能体规划方法中，公共信息要对所有智能体公开，而在本研究设计的基于动态智能体交互图的多智能体协同规划方法中，公共状态只需要向特定智能体公开，能够提高智能体的私有性并降低通信代价。例如，拍照状态的从属状态中包含姿态校准状态，而姿态校准状态显然属于姿态智能体而不属于相机智能体，则姿态校准状态是相机智能体中的公共状态。公共状态在规划中依据约束信息动态出现，当前姿态校准状态为公共状态不代表所有姿态校准状态都是公共状态。

显然，相机智能体不应该在局部规划中加入姿态校准这个状态，因为它属于姿态智能体。在生成公共状态的智能体中，它被视作一个不需要加入规划的“虚状态”，只有当虚状态被发送到正确的智能体，才成为需要加入规划的“实状态”。

本文提出动态智能体交互图作为智能体间信息交互的依据。

探测器的完整动态智能体交互图按子系统分解为多个子图，如图2所示。子图中的点按是否已经加入规划时间线分为已激活和未激活两部分，图中的点表示状态，有向边表示状态之间的关系。子图内部有向边从状态指向其从属状态，表示状态扩展过程及状态间从属关系。子图间的边从虚状态指向相应的实状态，表示智能体间状态发送过程。智能体交互图随规划进行动态变化，已激活状态扩展生成未激活状态，未激活状态中的公共状态被发送到其它智能体，未激活状态激活后转移到已激活区。图中实箭头表示状态扩展过程，虚箭头表示状态在图的不同区域中转移的过程。

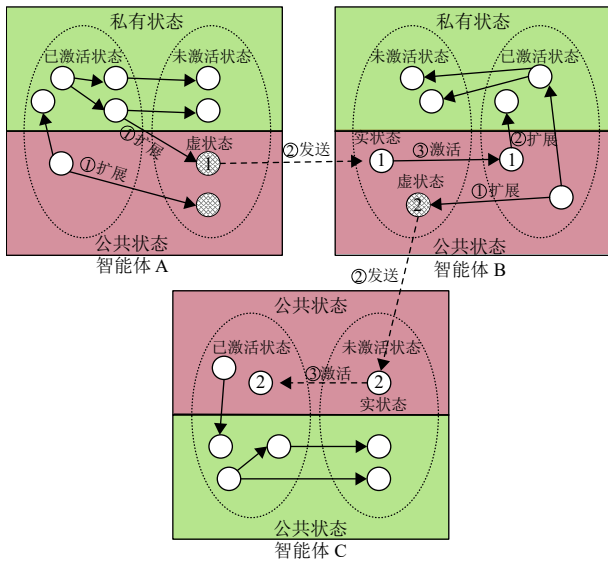


图2 动态智能体交互图

Fig. 2 Illustration of dynamic agent interaction graph

图2中, 智能体A中包含多个节点, 每个节点代表一个状态, 其中一个状态在规划中扩展后得到公共未激活状态1, 状态1在A中是一个虚状态, 状态1的信息被发送到智能体B中, 智能体B根据虚状态1的信息生成一个实状态1。实状态1在智能体B中被激活后按照规则继续扩展它的从属状态。完成发送过程后, 智能体A不再保留状态1。同理, B也会生成公共虚状态, 虚状态2由B中的规则引发, 但应该属于智能体C, 因此C会接收虚状态2的信息然后生成一个相应的实状态2, 并激活此状态继续规划。

2.2 多智能体数值约束处理

探测器任务规划考虑的数值约束主要包括时间约束和资源约束。

本研究采用多智能体时间约束网络建模智能体内部及智能体间的时间约束。时间约束网络中的节点代表状态的开始时刻或结束时刻, 网络中的边表示时间点之间的约束关系。时间点 a 和时间点 b 之间的时间约束关系用边 ab 和边 ba 表示, 边上的权值分别为 E_{ab} 和 E_{ba} , $b-a \leq E_{ab}$, $b-a \geq -E_{ba}$ ($E_{ab} \geq 0$, $E_{ba} \leq 0$), E_{ab} 表示 b 点在 a 点后最长时长, $-E_{ba}$ 表示 b 点在 a 点后最短时长。

智能体的状态之间的时间约束构成了当前智能体的简单时间网络^[15] (STN, Simple Temporal Network)。多智能体时间约束网络中包含每个智能体的局部STN和多个智能体STN之间的边。STN之间的边表示属于不同智能体的状态间的时间约束。

本研究通过基于图的最短路径算法的约束解耦方法实现智能体之间的时间约束处理。

当一个公共状态从虚状态转换为实状态时, 将其与主导状态之间的时间约束加入到主导状态所属智能体的STN当中, 采用Floyd-Warshall算法 (一种最短路径算法) 对此网络进行推理, 得到网络中所有边的最小值。公共状态的开始结束时刻都是网络中的节点, 这两个节点与时间零点间的边表示此状态的开始结束时刻的取值范围, 即 $t_s = [t_s^{\min}, t_s^{\max}]$ 、 $t_e = [t_e^{\min}, t_e^{\max}]$; 两个节点之间的边表示公共状态持续时长的取值范围, 即 $d = [d^{\min}, d^{\max}]$ 。至此, 公共状态将其与主导状态间的约束, 转换为其自身的时间约束信息, 实现了公共状态与其主导状态的时间约束解耦。

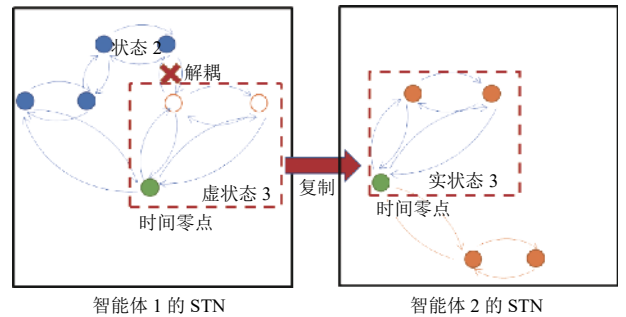


图3 多智能体时间约束网络解耦示意

Fig. 3 Illustration of multi-agent temporal network decoupling

如图3所示, 状态3是状态2的从属状态, 因此两个状态之间存在时间约束。当虚状态3发送到智能体2中成为实状态3时, 它与状态2之间的时间约束会被切断, 但是它与时间零点之间的约束和自身的持续时间约束会保留, 从而实现了两个STN之间的解耦。

对于资源约束, 提取所有影响资源量的状态, 构建资源约束网络。状态的开始结束时刻为网络中的节点, 资源量增加的点定义为生产点, 资源量减少的点定义为消耗点, 并增加一个源点和一个汇点。网络中的边分为3种: 内部边从执行时间晚的点指向执行时间早的点; 生产边从源点指向生产点; 消耗边从消耗点指向汇点。在资源约束网络中采用图的最大流算法判断资源量是否满足约束^[16]。本研究目前只考虑单个智能体的私有资源, 多智能体共享资源通过规划前定义每个智能体的用量上限转换为私有资源处理。

2.3 基于分布式求精搜索的规划空间规划

求精搜索是一种基于缺陷处理的规划方法, 由于

搜索空间中的节点是部分规划，因此是一种规划空间规划方法。在每一步规划中，算法从当前部分规划中选择并消解一个缺陷，从而产生一个新的部分规划。求精搜索考虑3种缺陷^[7]：

1) 未绑定变量：如果一个部分规划中的变量的值域不是唯一的，那么此变量就是一个未绑定变量。未绑定变量的消解方式是在值域中为此变量指定一个值。

2) 开放条件：一个未激活 (inactive) 状态就是一个开放条件。消解此缺陷的方式是融合 (merge)、激活 (activate) 或拒绝 (reject) 未激活的状态。

3) 威胁：一个状态加入部分规划后，可能会间接影响其他状态。比如，时间线上的状态不允许重叠，但新加入的状态可能会与时间线上已有状态重叠引发威胁缺陷。如果多个状态共享一个资源，也可能引起资源约束不一致的威胁。解决威胁的方式是在状态之间加入顺序约束。

图4所示为在多智能体规划过程中，每个智能体内部的规划流程。此流程与单智能体规划的主要区别在于公共状态缺陷处理和终止条件判断。

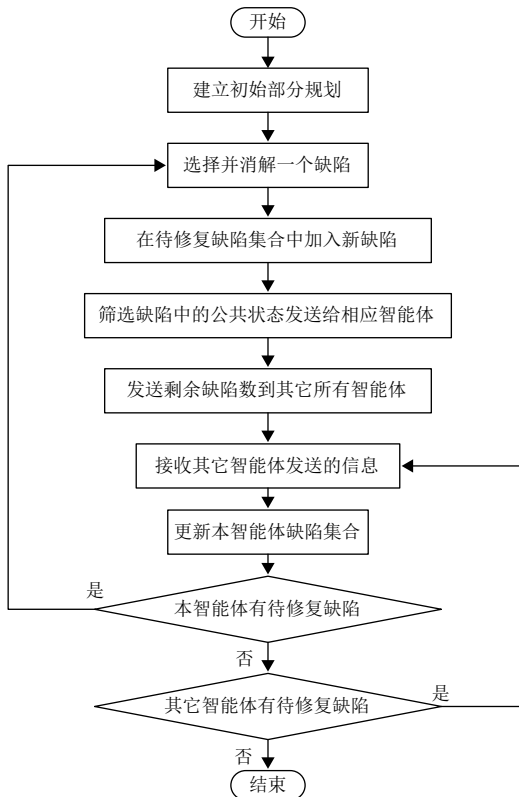


图 4 多智能体求精规划中每个智能体的规划流程
Fig. 4 Flowchart of multi-agent refinement planning

本研究的方法将共享资源转换为智能体私有资源处理，因此在3种缺陷中，只有开放条件缺陷会涉及多个智能体。开放条件缺陷就是一个未激活的状态，未激活状态可能是一个私有状态，也可能是公共状态。一个状态激活后，其从属状态就成为新的未激活状态。多智能体求精搜索需要解决的主要问题就是如何处理未激活的公共状态。

本研究不设置多智能体协调中心，而是基于动态智能体交互图定义的状态扩展和转移规则处理未激活公共状态。将未激活的公共状态发送到相应子系统的智能体中，成为该智能体的未激活状态，实现跨智能体的缺陷转移。第2.2节的时间约束传播及解耦方法给出了公共状态传递过程中时间约束的处理方式。在多智能体求精搜索算法中，为了实现公共状态从智能体B到A的传递，需要以下具体步骤：

1) 第*i*步规划中，智能体B记录新生成的开放条件缺陷，加入集合 Op_i ；在第*i*步规划后，从 Op_i 中筛选出所有公共状态。

2) 获取每个公共状态*T*的谓词名称，包括智能体名称（即子系统名称，如“AttitudeSys”），状态类型（如“AttitudeSys.Pointing”），状态名称（如“pointing”）和时间信息。时间信息包括该状态的开始时刻*s*、结束时刻*e*和持续时长*d*三种时间变量的可行区间，即 $s \in [s_{min}, s_{max}]$ ， $e \in [e_{min}, e_{max}]$ ， $d \in [d_{min}, d_{max}]$ 。

3) 智能体B将公共状态的谓词名称和时间信息发送到智能体A中，并从B的候选缺陷中剔除公共状态。

4) 智能体A生成与公共状态智能体名称、状态类型相同的新状态 T_{new} 。为了避免与已有状态重名，在公共状态的名称上添加元素，构造新的状态名称，比如在pointing的基础上增加单词“add”和表示规划轮数的数字*n*。

5) 智能体A按照接收到的公共状态的时间信息，为新生成的状态 T_{new} 添加时间约束，将开始结束时刻和状态持续时长限制在规定的值域内。在时间约束网络中加入 T_{new} ，进行约束传播。

6) 智能体A将接收到的状态加入开放条件候选缺陷集合中。

这样一来，未激活的公共状态从B转移到A中，由A负责修复。

多智能体规划的结束条件与单智能体规划不同。单智能体规划只需要考虑本地的缺陷是否都得到修

复, 规划步数和时长是否达到上限。而多智能体规划中的每个智能体在规划步数和时长达到用户设置的上限之前, 不仅需要以自身是否有待处理的缺陷, 也需要考虑其他智能体是否有待处理缺陷。因为对于一个暂时没有待处理缺陷的智能体, 其它智能体可能会在后续规划中向它发送公共缺陷。只有确认所有智能体都不需要继续处理缺陷, 才能停止规划。

3 仿真实验及结果分析

为了验证方法的有效性, 本研究在火星环绕器领域构建测试用例, 考虑载荷相机、姿态系统、对地通信等9个子系统(见表1), 规划程序使用C++语言实现。

表1 火星环绕器各子系统功能

Table 1 Functions of subsystems in Mars Orbiter

子系统	功能	约束中定义的公共状态数
姿态子系统	控制环绕器姿态	0
载荷相机	拍照	1
对地通信系统	与地球通信	1
对火通信系统	与巡视器通信	2
推进系统	推进轨道器	0
太阳帆板	获取太阳能	2
GNC模式调整	调整导航模式	3
轨道控制	调整轨道	1
综合电子系统	调整测控模式	1

多智能体规划方法将环绕器的各子系统抽象为智能体, 共有9个智能体, 各智能体的状态之间相互耦合, 共有37个状态(见表2)和49个约束, 约束中定义的公共状态有11个。其中载荷相机中包含载荷电池, 用于验证资源约束处理能力。

表2 火星环绕器各子系统状态列表

Table 2 States of subsystems in Mars Orbiter

子系统	状态列表
姿态子系统	指向、转动
载荷相机	关、打开、开、对准、工作、关闭
对地通信系统	关、打开、开、打开通信、通信、关闭
对火通信系统	关、打开、开、打开通信、通信、关闭
推进系统	关、打开、开、开始推进、推进、关闭
太阳帆板	保持、对日、锁定
GNC模式调整	默认模式、模式1、模式2
轨道控制	空闲、调整轨道
综合电子系统	默认模式、模式1、模式2

各智能体间有如下约束关系。载荷相机拍照需要姿态系统指向目标; 对地通信需要姿态系统指向地球; 对火通信需要姿态指向巡视器, 对火通信后需要对地通信; 太阳帆板充电需要姿态系统指向太阳, 太阳帆板锁定前需要综电系统调整测控为模式1; GNC模式调整系统有两种模式, 模式1前需要太阳帆板锁定, 模式1中需要姿态转动, 模式2前需要太阳帆板对日; 轨道控制系统调整轨道的同时需要推进系统工作; 综合电子系统有两种测控模式, 进入模式2前需要GNC模式调整系统进入模式2。

设计5个测试问题, 测试问题中每个子系统的规划目标数量由1到5递增, 总目标数由9到45递增。分别采用提出的多智能体规划方法和EUROPA的进行规划, 对比规划用时和规划步数。每个问题进行10次测试, 统计规划用时的平均值。

对多智能体规划空间规划方法(MAPP)进行串行(MAPP-串行)和并行(MAPP-并行)两种模式的测试。多智能体串行模式指在每一轮规划中, 多个智能体依次规划一步, 同一时刻只有一个智能体处于规划状态, 一轮规划后智能体间交互信息, 再进行下一轮规划。并行模式指在一轮规划中多个智能体同时规划一步, 交互信息后再进行下一轮规划。两种模式的算法本质相同, 每一步规划中各智能体在逻辑上都是并行的, 不受其它智能体影响。串行模式主要用于说明本文提出的方法即使不利用并行计算也能够提高规划效率。

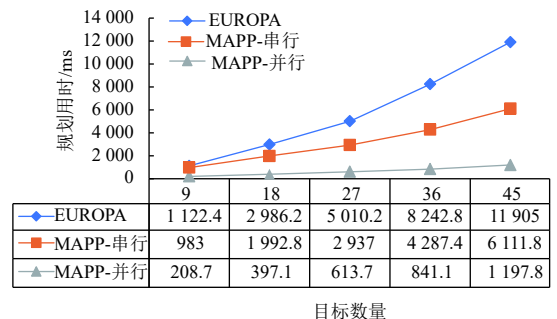


图5 规划用时结果曲线

Fig. 5 Computational results of runtime curves

图6展示了采用本文提出的多智能体规划方法得到的规划结果序列, 序列中的空白部分表示没有新动作指令, 探测器子系统保持上一个动作结束后的状态。

输入问题中，探测器每个子系统有一个规划目标状态。规划结果显示姿态、载荷、对地通信、推进、太阳帆板、GNC模式调整及综电系统的规划结果中实现了不止一个状态，说明算法成功实现了智能体间的协

同规划，除了自身问题文件中的目标状态，智能体还能配合其他智能体处理公共状态。不同系统的状态间的时间关系也满足前文所述的智能体间约束。图中仅标出了智能体本身的目标状态和公共状态。

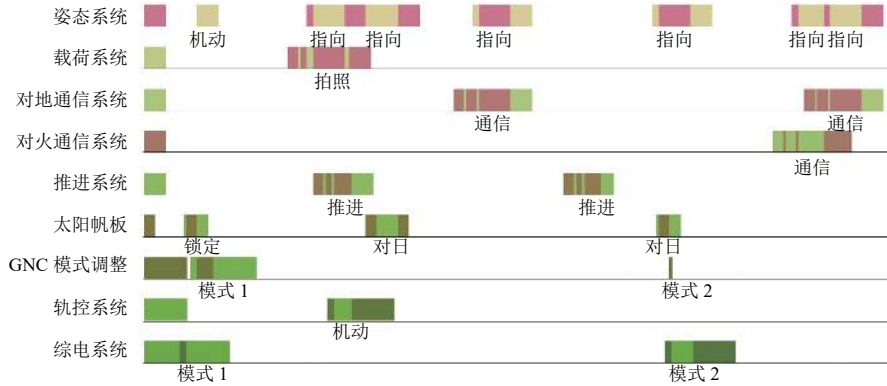


图 6 各系统规划结果甘特图

Fig. 6 Gantt chart of subsystems

3种规划方法的规划用时都随着问题的目标数增加而增长，MAPP-并行用时最少，EUROPA用时最多。多智能体规划空间规划的串行模式比EUROPA的规划用时少，说明通过将子系统抽象为智能体，分割搜索空间，能够提高规划效率。

多智能体规划空间规划中，每个智能体只需要选择并修复局部缺陷，而不必比较所有子系统中的待处理缺陷，降低了缺陷处理的复杂程度，提高了规划效率。

多智能体规划提高了时间约束处理的效率。假设智能体*i*的时间约束网络中的节点数为 k_i ， n 个智能体的时间约束网络中的节点数之和为 $K = \sum_{i=1}^n k_i$ 。采用Floyd-Warshall算法对时间约束网络进行推理，此算法的时间复杂对为 $O(k^3)$ ， k 为网络中的节点数。对于多智能体时间约束网络，每个智能体的时间约束推理的时间复杂度为 $O(k_i^3)$ ，多个智能体串行推理的时间复杂度为 $O(\sum_{i=1}^n k_i^3)$ 。在集中式规划中，所有智能体的节点在同一张时间约束网中，约束推理的时间复杂度为 $O(K^3) = O((\sum_{i=1}^n k_i)^3)$ 。显然将时间约束网络分解后的推理效率更高。

并行模式的多智能体规划空间规划中，规划步数为所有智能体的规划步数中的最大值。与EUROPA和串行模式的多智能体规划相比，并行模式的多智能体规划增加了参与规划的计算资源，规划步数更少，规

划用时更短，极大提高规划效率。规划步数对比参见图7。

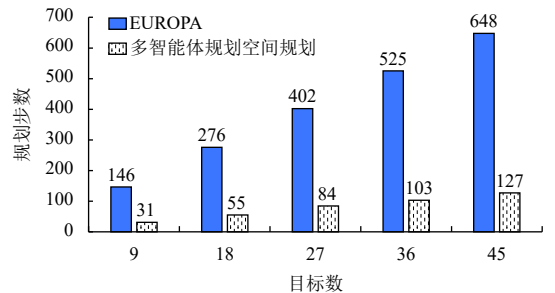


图 7 规划步数结果对比

Fig. 7 Comparison of planning steps

对于子系统众多、领域知识复杂、任务目标多的深空探测器任务规划问题，适于采用多智能体规划空间规划。计算资源有限时可采用串行模式，多个智能体轮流规划，与将多子系统视为一个整体的单智能体规划空间规划相比，可实现更高的规划效率；计算资源充足则可采用并行模式，如多进程并行或多计算机并行计算，进一步缩短规划时间。

4 结 论

本研究实现了基于求精搜索的深空探测器多智能体规划空间规划的理论设计和仿真验证。结合数值约束处理技术和多智能体规划方法，即满足了深空探测

器的时间资源约束处理需求,又实现了高效的多智能体协同规划。实验证明所设计的规划方法能够为深空探测器任务规划生成合理规划,并显著提高了规划效率,为未来深空探测任务规划提供了一种新的可行方案。

参 考 文 献

- [1] BARREIRO J, BOYCE M, DO M, et al. EUROPA: a platform for AI planning, scheduling, constraint programming, and optimization[Z]. ICAPS-12 International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS). [S. l.]: ICKEPS, 2012.
- [2] FRANK J, JONSSON A. Constraint-based attribute and interval planning[J]. *Journal of Constraints Special Issue on Constraints and Planning*, 2003(8): 339-364.
- [3] MUSCETTOLA N, NAYAK P P, PELL B, et al. Remote agent: to boldly go where no AI system has gone before[J]. *Artificial Intelligence*, 1998(103): 5-47.
- [4] MUSCETTOLA N, FRY C, RAJAN K, et al. On-board planning for New Millennium Deep Space One autonomy[C]//*IEEE Aerospace Conference*. [S. l.]: IEEE, 2002.
- [5] TRAN D, CHIEN S, SHERWOOD R, et al. The autonomous sciencecraft experiment onboard the eo-1 spacecraft[C]//*Autonomous Agents and Multiagent Systems*, 2004. [S. l.]: IEEE, 2004.
- [6] CHIEN S, RABIDEAU G, KNIGHT R, et al. ASPEN - automating space mission operations using automated planning and scheduling[C]//*In Proceedings of SpaceOps*. Toulouse, France: [s. n.], 2000.
- [7] KNIGHT R, RABIDEAU G, CHIEN S, et al. Casper: space exploration through continuous planning[J]. *IEEE Intelligent Systems and Their Applications*, 2001, 16(5): 70-75.
- [8] 金颢, 徐瑞, 崔平远, 等. 基于状态转移图的启发式深空探测器任务规划方法[J]. *深空探测学报(中英文)*, 2019, 6(4): 364-368.
JIN H, XU R, CUI P Y, et al. Heuristic search based on state transition graphs for deep space task planning[J]. *Journal of Deep Space Exploration*, 2019, 6(4): 364-368.
- [9] 王晓晖, 李爽. 深空探测器约束简化与任务规划方法研究[J]. *宇航学报*, 2016, 37(7): 768-774.
- WANG X H, LI S. Research on constraint simplification and mission planning method for deep space explorer[J]. *Journal of Astronautics*, 2016, 37(7): 768-774.
- [10] BRAFMAN R I, DOMSHLAK C. From one to many: planning for loosely coupled multi-agent systems[C]// ICAPS. [S. l.]: ICAPS, 2008.
- [11] ŠTOLBA M, KOMENDA A. The MADLA planner: multi-agent planning by combination of distributed and local heuristic search[J]. *Artificial Intelligence*, 2017(252): 175-210.
- [12] TORREÑO A, ONAINDIA E, SAPENA Ó. FMAP: distributed cooperative multi-agent planning[J]. *Applied Intelligence*, 2015(41): 606-626.
- [13] TORREÑO A, ONAINDIA E, KOMENDA A, et al. Cooperative multi-agent planning: a survey[J]. *ACM Computing Surveys*, 2017(50): 1-32.
- [14] XU R, CUI P Y, XU X. Realization of multi-agent planning system for autonomous spacecraft[J]. *Advances in Engineering Software*, 2005(36): 266-72.
- [15] DECHTER R, MEIRI I, PEARL J. Temporal constraint networks[J]. *Artificial Intelligence*, 1991(49): 61-95.
- [16] 陈德相, 徐瑞, 崔平远. 航天器资源约束的时间拓扑排序处理方法[J]. *宇航学报*, 2014, 35(6): 669-676.
CHEN D X, XU R, CUI P Y. A Temporal topological sort processing method for spacecraft resources constraints[J]. *Journal of Astronautics*, 2014, 35(6): 669-676.
- [17] IATAURO M. How does EUROPA solve problems?[EB/OL](2014-10-22)[2021-04-06].<https://github.com/nasa/europa/wiki/Bak-Problem-Solving>, (accessed 01.10.20).

作者简介:

赵宇庭(1994-),女,博士生,主要研究方向:航天器自主任务规划,多智能体规划技术。

通讯地址:北京理工大学宇航学院22信箱(100081)

电话:(010)68913550

E-mail: zhaoyuting_bit@163.com

徐瑞(1975-),男,教授,主要研究方向:航天器自主任务规划、自主导航、智能控制。

通讯地址:北京理工大学宇航学院22信箱(100081)

电话:(010)68913550

E-mail: xurui@bit.edu.cn

Mission Planning Based on Dynamic Agent Interaction Graph for Deep Space Probes

ZHAO Yuting^{1,2}, XU Rui^{1,2}, LI Zhaoyu^{1,2}, ZHU Shengying^{1,2}

(1. School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China; 2. Key Laboratory of Autonomous Navigation and Control for Deep Space Exploration, Ministry of Industry and Information Technology, Beijing 100081, China)

Abstract: Facing the increasingly complex deep-space exploration missions and the dynamic space environment, deep-space probes need efficient planning methods for the fast generation of plans. The distribution and concurrency of subsystems make a probe suitable to be modeled as a multi-agent system. Existing multi-agent planners, however, cannot be used directly in mission planning of deep space probes that involve handling numeric constraints such as time resources. To solve the above problem, a multi-agent mission plan-space planning method based on distributed refinement search was proposed. A dynamic agent interaction graph (DAIG) was designed to coordinate interactions between agents during planning. Temporal constraints and resource constraints were modeled as constraint satisfaction problems and were handled by graph theory methods. Experiments show that the method proposed in this paper can save computing time of mission planning problems for a probe with multiple subsystems.

Keywords: deep space probe; multi-agent plan-space planning; numeric constraints

Highlights:

- A distributed multi-agent plan-space planning method was developed for deep-space probe planning.
- A dynamic agent interaction graph was proposed to coordinate the multi-agent interactions during the planning process.
- Temporal constraints between agents were handled by time constraints propagation between temporal networks.
- The proposed multi-agent planning method can plan for a probe correctly and save computing time.

[责任编辑：高莎，英文审校：宋利辉]