

基于状态差异的火星巡视器快速任务规划修复方法

陈超^{1,2}, 徐瑞^{1,2}, 李朝玉^{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: 火星环境存在不确定、电子设备故障难预测等特点, 严重影响巡视器已有规划在火星表面实际应用的效果, 执行任务易造成失败。在火星巡视器与地面站之间通信存在长时延的客观事实下, 提出一种基于状态差异的快速任务规划修复方法。利用感知状态与动作执行的必要状态之间的差异, 从已有规划中提取关键信息来构建不同时刻的部分状态, 提出了基于部分状态的火星巡视器快速任务规划修复策略, 并考虑实际状态与部分状态之间的差异, 设计了基于状态差异的搜索空间删减方法, 有目的地生成、扩展节点并消解冲突, 以提高规划修复效率。仿真实验结果表明, 该方法不仅提高了火星巡视器任务规划修复的效率, 而且保障了规划稳定性, 可为航天器快速自主应对执行时的突发事件提供技术支持。

关键词: 规划修复; 火星巡视器; 部分状态; 状态差异

中图分类号: V419+9

文献标识码: A

文章编号: 2096-9287(2021)02-0124-08

DOI: 10.15982/j.issn.2096-9287.2021.20200075

引用格式: 陈超, 徐瑞, 李朝玉. 基于状态差异的火星巡视器快速任务规划修复方法[J]. 深空探测学报(中英文), 2021, 8(2): 124-131.

Reference format: CHEN C, XU R, LI Z Y. Fast mission plan repair method for Mars rover based on state difference[J]. Journal of Deep Space Exploration, 2021, 8(2): 124-131.

引言

由于距离地球遥远, 火星探测器与地面站之间的通信存在3~20 min的长时延, 严重影响着任务的成败。火星探测器需具备高度的自主能力, 例如高增益天线的自主指向、利用星敏进行姿态校准、在线数据的存储与转发, 以及在时间和资源约束条件下活动的合理安排等^[1], 以降低地面依赖性、缩短任务周期。自主任务规划技术可实现这一目标^[2]。现有文献已经从动作^[3]、资源^[4]、时间线^[5]等多个角度对自主任务规划技术进行研究。但目前人类对火星的认知不全面, 火星表面地形的变化、土壤的坚硬程度、岩石的大小等仍存在极大的不确定性, 影响着巡视器执行任务的时间长短、消耗能量以及存储空间大小等, 严重制约着既定规划的成功执行。例如, “探路者号”(Pathfinder)任务中的“索杰纳号”(Sojourner)巡视器就因为先在先验规划过程中没有充分考虑到这些不确定性因素的影响, 整个任务中大约有70%的时间处于闲置状态^[6]。此外, 风沙等突发事件、电子器件故障等意外事件, 都会对任务的成功执行造成巨大威胁。因此, 火星巡视器不仅需要具备任务规划能力, 自主管理日常活动,

更需要具备任务规划修复能力, 自主应对执行过程中出现的执行失败情况, 以增强自身的鲁棒性, 提高任务回报率。

在应对突发事件导致任务规划执行失败的问题方面, 存在两种策略: 完全重规划和规划修复^[7]。其中, 完全重规划是指放弃已有的规划结果, 重新决策出一个新的动作序列来完成目标, 而规划修复是指通过增、删、替换、移动等操作, 对已有规划结果进行修补来达成任务目标。虽然规划修复在理论上不一定比完全重规划简单^[8], 但大量的仿真实验表明规划修复效率更高^[9-12]。Scala等^[13]基于巡视器动作模式的多样性, 采用约束满足问题(Constraint Satisfaction Problem, CSP)求解技术, 在规划执行失败时, 通过动作模式重构完成任务规划修复。但是该方法不改变动作顺序及类型, 无法处理一般的规划执行失败问题。Gallien等^[14]将执行阶段设计成感知-修复-行动三阶段的循环过程, 对执行任务过程中出现的错误, 在限定时间内采用规划器推理得出相应的解, 用于解决执行失败的问题。但是该方法采用偏序规划方法, 效率较低。Guzman等^[15]提出响应式规划(Reactive Plan, RP)方法, 为执行体设计了一个响应式规划

收稿日期: 2020-11-19 修回日期: 2020-12-23

基金项目: 国家重点研发资助项目(2019YFA0706500); 国家自然科学基金资助项目(61976020)

器，通过与任务规划器互相配合，并采用修复结构树来提高规划修复的速度。但是修复结构树的长度和深度需要通过机器学习获得，不能普遍适用于所有情况；并且即使修复结构树只包含少量动作，其中也会包含大量无用节点，这样会造成存储空间和搜索时间的浪费。

针对巡视器在火星表面执行任务过程中由环境不确定性和自身设备故障等引起的规划执行失败问题，本文提出了一种基于状态差异的规划修复方法（State Difference Plan Repair, RPDS），使巡视器能够迅速从失败中恢复正常。与响应式规划方法RP不同的是，本文不提前构建修复结构树以辅助规划修复，而是在执行失败后，利用不同状态之间的差异，有目的地生成搜索节点，并对扩展的节点根据谓词冲突进行筛选，从而删减搜索空间，加快巡视器规划修复的速度。

本文的结构组织如下：第一部分构建火星巡视器任务模型并引出部分状态的概念，为后续的方法描述奠定基础。第二部分提出基于部分状态的火星巡视器快速任务规划修复策略，为第三部分描述基于实际状态与部分状态之间的差异而设计的搜索空间删减方法建好框架。第四部分给出仿真测试结果和分析。最后给出本文结论。

1 火星巡视器任务模型

自主任务规划属于高层次的智能决策，在给定初始状态下找到满足约束并且能够完成任务目标的动作序列的一种技术^[16]。本文假设火星巡视器的姿态、导航等系统具备一定自主能力，在接收各自的任务指令后，各系统能够以一定的方法完成指令。由此，定义如下几个方面的规划修复问题。

定义1 规划修复问题 一个规划修复问题由变量的值域 V 、动作集合 A 、初始状态 I 和目标状态 G 4个元素组成，即 $P = (V, A, I, G)$ 。规划修复问题就是如何在变量的取值范围内，考虑动作的组合方式，通过搜索等方式找到由初始状态到达目标状态的动作序列。

采用PDDL2.1^[17]（Planning Domain Definition Language 2.1）描述上述规划修复问题。PDDL2.1采用一阶谓词逻辑表示知识，例如 $at(rovers, t)$ 表示巡视器 $rovers$ 现在位于地点 t 处。值域 V 包含所有谓词所有可能的取值。所有谓词的一次赋值构成一个状态。初始状态 I 描述规划开始时巡视器的位置、指向、载荷开关机状态等，目标状态 G 描述科学任务目标，例如对某地点进行采样、拍照、传输数据等。状态之间的转换

需要动作 A 的支持。巡视器的动作包括导航、太阳能帆板充电、采样、照相、数传等，如表1所示。动作由前提条件和效果构成，而效果又包括添加效果和删除效果，可以表示成 $a = (p_{res}, e^+, e^-)$ ，如图1所示。如果在当前状态 s 下，动作 a 的前提条件得到满足，则执行动作 a 后，状态变为 $s' = s \setminus a.e^- \cup a.e^+$ 。

表1 火星巡视器模型中的活动及其含义

Table 1 Activities in the Mars rover model and their meanings

活动名称	含义
navigate	导航
recharge	充电
sample_soil	土壤采样分析
sample_rock	岩石采样分析
drop	丢弃样本
calibrate	校准相机
take_image	成像
communicate_soil_data	向着陆器传输土壤分析数据
communicate_rock_data	向着陆器传输岩石分析数据
communicate_image_data	向着陆器传输图像数据

```
(:durative-action sample_rock
:parameters (?x - rover ?s - store ?p - waypoint)
:duration (= ?duration 8)
:condition (and (over all (at ?x ?p)) (at start (at ?x ?p))
(at start (>= (energy ?x) 5))
(at start (at_rock_sample ?p))
(at start (equipped_for_rock_analysis ?x))
(at start (store_of ?s ?x))
(at start (empty ?s))
)
:effect (and (at start (not (empty ?s))) (at end (full ?s))
(at end (have_rock_analysis ?x ?p))
(at start (decrease (energy ?x) 5))
(at end (not (at_rock_sample ?p)))
)
)
```

图1 对岩石采样的动作模型示例

Fig. 1 An example of an action model for rock sampling

一般情况下，火星巡视器接收任务指令，进行任务规划，在获得规划结果后执行具体指令时，需要传感器感知环境信息并将其传回监控系统，用于观察和控制任务的执行情况。然而感知状态中含有许多与具体动作执行无关的冗余信息，影响任务规划修复的效率。例如，对于原位采样动作，感知状态中着陆器的位置、通信设备的占用情况等就不必要显示。因此，利用感知状态与动作执行的必要状态之间的差异，定义部分状态，在状态中直接提取与动作直接相关的信息，以减小任务规划修复的搜索空间大小，加快任务规划修复速度。

定义2 部分状态 若状态 s_0 经过动作 a 作用后变为状态 s_1 ，那么 s_0 下的部分状态 G 应该包含动作 a 的前提条

件, 通过状态回退函数 $\Gamma(s_1, a)$ 计算得到, 即 $G = \Gamma(s_1, a) = s_1 \setminus a.e^+ \cup a.e^-$ 。特别地, 对于一个规划结果, 即动作序列 $\Pi_A = \langle a_1, \dots, a_n \rangle$, 若从目标状态开始应用状态回退函数, 则可以获得部分状态序列 $\langle G_0, G_1, \dots, G_n \rangle$, 满足 G_n 是目标状态, $G_0 \subseteq I$, $G_{i-1} = \Gamma(G_i, a_i)$ 。此时, 部分状态 G_i 包含动作序列 $\langle a_{i+1}, a_{i+2}, \dots, a_n \rangle$ 中所有动作的前提条件。

假定任务目标是对 w_2 处的土壤进行采样分析并将

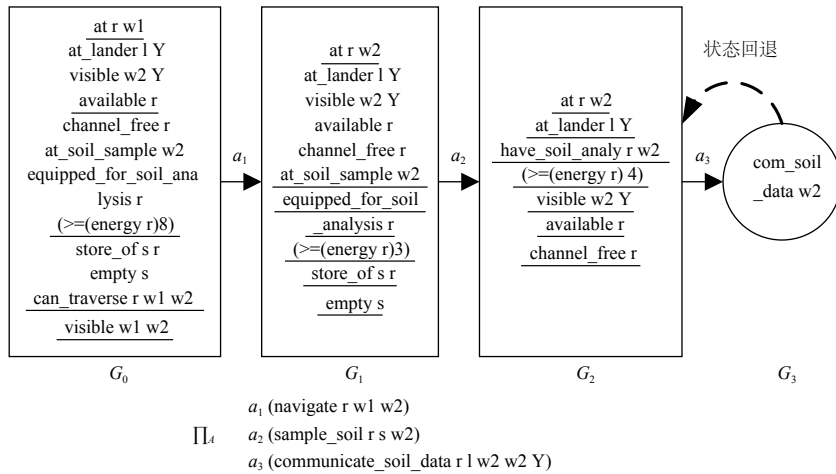


图2 规划 Π_A 的部分状态序列
Fig. 2 Partial state sequence of the plan Π_A

2 火星巡视器快速任务规划修复策略

在任务执行的过程中, 火星巡视器将要执行动作的前提条件可能会因为环境的不确定性或者自身设备的故障而不再得到满足, 执行效果达不到预期, 从而被迫中止任务。任务规划修复就是在尽可能保留原有规划结果的前提下, 运用一定技术和方法, 使巡视器能够从失败状态中恢复正常、继续执行任务。相较于感知状态而言, 部分状态包含未执行动作的必要条件, 不含冗余信息, 缩小了状态空间的规模, 有利于快速完成任务规划修复。因此, 设计基于部分状态的火星巡视器快速任务规划修复策略。

如图3所示, 在执行动作 a_{i+1} 时, 若由于某种原因, a_{i+1} 的前提条件没有得到满足, 巡视器的状态由理想状态 G_i 变成了 G'_i 。基于部分状态的规划修复策略尝试找到一条路径, 使得巡视器能够从失败状态 G'_i 转换到已有规划中的某个部分状态。具体来说, 在动作 a_i 执行失败时, 假设失败时的状态为 S , 从失败动作的前提条件所在的部分状态 G_{i-1} 开始, 通过智能搜索寻找一条路径到达下一个状态 G' , 若 $G' \subseteq S$, 说明找到了解, 此时, 提取相应的动作序列即可完成任务修复;

分析结果传回着陆器, 对应的动作序列 Π_A 及部分状态序列如图2所示。 G_0 表示初始部分状态, G_3 为目标部分状态, 图2中带下划线的谓词为下一个动作的前提条件。每个状态下的所有动作前提集合即构成一个部分状态, 为规划修复提供可修修复目标。此时, 若外界环境的发生改变或其它原因导致待执行动作的前提条件得不到满足, 便可利用部分状态中来设计快速任务规划修复策略。

否则, 继续通过搜索路径转移到下一个状态, 直到找到解, 或者达到限制条件, 例如修复时间要求、搜索深度要求等。

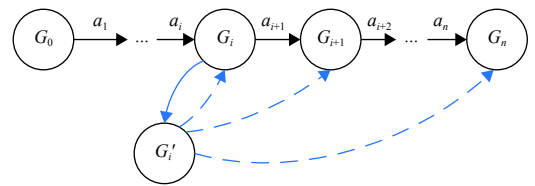


图3 任务规划修复策略示意图
Fig. 3 Illustration of mission plan repair strategy

火星巡视器快速任务规划修复策略的伪代码如图4所示。第1~4行对规划结果中的动作序列运用状态回退函数, 通过计算得到部分状态集合 G 。以 G 为可选修复目标, 引导修复操作。当巡视器监控到动作 a_k 的前提条件被破坏而难以执行时, 记此时的巡视器状态为 S (第5行), 未执行动作集合为 P_A (第6行)。其中, P_A 既是原规划结果中的动作序列, 又是一个不完整的规划修复解。规划修复的要求之一就是尽可能地保留 P_A 中的元素。第8~28行是整个策略的核心部分。从失败动作前提所在的部分状态 G_{k-1} 开始, 尝试寻找

使 P_A 一致的修复动作序列，即 P_A 在当前状态 S 下能够得到应用。若不存在这样的动作序列，换到下一个部分状态，重复该过程。

算法 1: 基于部分状态的火星巡视器快速任务规划修复策略

输入: 规划结果 A
输出: 规划修复解 P_A 或 failure

1. $G_n \leftarrow$ mission goals
2. **for** $i = n$ to 1
3. $G_{i-1} = \Gamma(G_i, a) // a$ is in A
4. **end for**
5. $S \leftarrow$ the state of the rover when action a_k fails executing
6. $P_A \leftarrow \{a_k, a_{k+1}, a_{k+2}, \dots, a_n\}$
7. $findsolution \leftarrow$ false
8. **for** $i = k-1$ to $n-1$
9. $G_{Temp} \leftarrow G_i$
10. $T_{ree} \leftarrow G_{Temp}$
11. **if** $G_{Temp} \subseteq S$
12. $P_A \leftarrow P_A \setminus \{a_{i+1}, a_{i+2}, \dots, a_n\}$, $findsolution \leftarrow$ true
13. **return** P_A
14. **else**
15. **while** $T_{ree}.depth < n-k+4$
16. **for** $\{a | a \in A, \text{applicable for repair in } G_{Temp}\}$
17. $G' = \Gamma(G_{Temp}, a)$
18. **if** there are no conflicts in G' and T_{ree}
19. $T_{ree} \leftarrow T_{ree} \cup G'$
20. **if** $G' \subseteq S$
21. $P_A \leftarrow P_A \cup \{\text{actions from } G' \text{ to } G_{Temp}\} \setminus \{a_{i+1}, a_{i+2}, \dots, a_n\}$, $findsolution \leftarrow$ true
22. **return** P_A
23. **end if**
24. **end if**
25. **end for**
26. $G_{Temp} \leftarrow \{g | g \text{ is an unused node in } T_{ree}\}$
27. **end while**
28. **end for**
29. **if** $findsolution$ is false
30. **return** failure

图4 火星巡视器快速任务规划修复策略伪代码

Fig. 4 Pseudocode of fast mission plan repair strategy for Mars rover

在每次求解的过程中，构造一颗搜索树 T_{ree} 以存储中间状态信息。 T_{ree} 中的节点是部分状态，边是动作。其节点的生成和扩展通过回退函数完成，直至达到限制要求。图4中的策略采用宽度优先原则对树进行扩展，并对树的生长过程进行了限制，即树的深度需要小于可能扩展的部分状态数量+3（第15行，从第8行可知 i 的取值跨度为 $n-k+1$ ，则可能扩展的部分状态数量也为 $n-k+1$ ，深度限制加3即为第15行中的 $n-k+4$ ）。通过限制树的深度，避免树的无穷扩展，缩小算法的搜索时间，以实现快速完成修复的目的。当扩展的节点在 S 下被满足时，通过节点和边的关系逆序追溯修复动

作序列，对 P_A 进行修补（第21行），返回修复解。若所有尝试都失败了，策略返回失败标志，通知巡视器调用规划器进行完全重规划或等待地面救援。

3 基于状态差异的搜索空间删减方法

在实际应用火星巡视器快速任务规划修复策略时，节点需要频繁地生成、扩展（第17~19行）。此时，若使用盲目搜索，无疑会增加问题求解的时间。相反，若能充分利用实际状态与部分状态之间的差异，设计相应的节点生成方法和冲突消解方法，有目的的生成和扩展节点，会减小搜索空间大小，从而缩短搜索的时间，提高方法的效率。

3.1 节点生成方法

在PDDL2.1的描述中，状态主要由谓词表示。当动作执行失败时，状态中只有部分谓词发生变化。因此，针对执行失败的动作，结合失败时的实际状态与部分状态之间谓词的差异，如图5所示，将部分状态分割成相同谓词和相异谓词两部分。修复时，忽略相同谓词，采用状态回退函数对相异谓词回退，进行后向搜索，有目的地扩展并生成搜索节点，可以减少扩展节点的总数，加快搜索效率。而在节点扩展的过程中，以相异谓词为起点进行回退后的状态可能会删除相同谓词部分，导致最终得到的修复解无法满足期望要求，从而诱发新一轮的搜索过程，降低修复效率。因此，在得到修复解之后，需要检查解的可行性：即在感知状态下模拟应用规划修复解后，判断推导的理论状态是否包含相应的部分状态，若包含，说明修复解可行，否则应该舍弃该节点继续处理下一个节点。

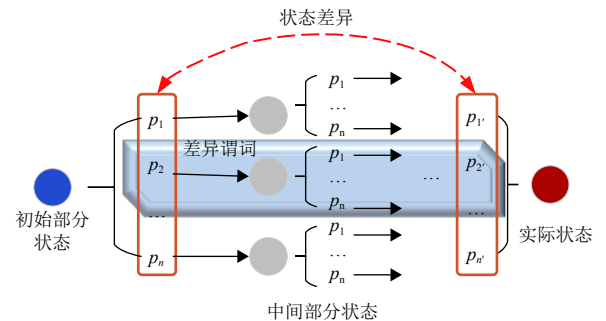


图5 基于状态差异的节点生成方法示意

Fig. 5 Node generation method based on state difference

3.2 冲突消解方法

一个部分状态由多个谓词组成，而每个谓词都是某一个未执行动作的前提条件，因此在每次通过状态回退扩展节点的时候，存在多个候选动作可用于回退。当选择其中一个进行扩展时，该部分状态 G 可被

分为动作的效果集合 $e_{\text{eff}}(a)$ 和其它部分谓词集合 o_{thers} 两部分。相应地,由该动作回退得到的部分状态 G' 也可被分为两部分,即 o_{thers} 以及前提条件 $p_{\text{re}}(a)$ 。但 o_{thers} 和 $p_{\text{re}}(a)$ 之间可能存在冲突。如图6所示,部分状态 $G = \{at(r, w_1), have_image(r, O_1, color), \dots\}$,此时,存在回退动作 $take_image(p_{\text{re}}: at(r, p), \dots; e_{\text{eff}}: have_image(r, O_1, color), \dots)$,则回退状态 $G' = \Gamma G, take_image = \{at(r, w_1), at(r, p), \dots\}$ 。然而巡视器在某一时刻只能位于一个位置,因此 G 和 G' 之间存在冲突。

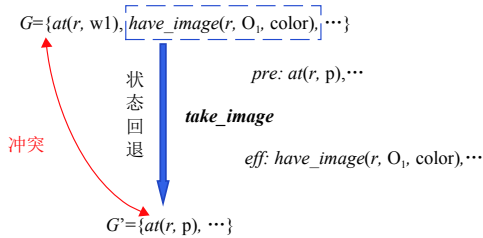


图6 部分状态与回退状态之间的冲突示意

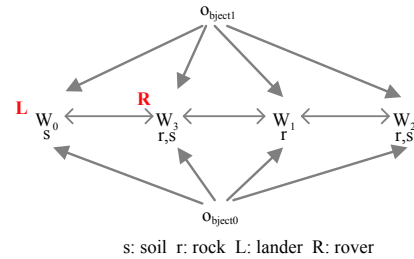
Fig. 6 Conflict between the partial state and the regressed state

通过分析可以发现,发生冲突的部分状态节点至少有两个名称相同但参数不相同的谓词成分,如 $at(r, w_1)$ 与 $at(r, p)$ 。但名称相同而参数不同的谓词不一定构成冲突,例如描述巡视器载荷能力的谓词($supports?camera?mode$),用于表征相机具有的功能模式,例如光谱成像、立体成像、红外成像等,这类谓词可以在同一个部分状态中并存而不产生冲突。因此,有必要针对火星巡视器,明确会引发部分状态节点冲突的谓词。在第1部分建立的火星巡视器任务模型中,易产生冲突的谓词是($at?rover?waypoint$)和($at_lander?lander?waypoint$),分别描述巡视器和着陆器的位置。由此,可以设计节点中的冲突消解方法如下:在判断节点是否存在冲突时,首先判断部分状态中是否存在可能引发冲突的谓词;若存在,再进一步判断其参数是否会引发冲突。冲突的部分状态自身就存在矛盾,所以在根据回退动作生成节点后,可以将自身冲突的节点删除,以减少要扩展的节点,进一步节省问题求解的时间。

4 仿真测试

为验证本文提出的基于状态差异的火星巡视器快速任务规划修复方法RPDS的有效性,设置如图7所示的仿真场景:存在4个地点 w_0, w_1, w_2, w_3 ,各地存在岩石或者土壤样本; w_0 和 w_3 之间、 w_3 和 w_1 之间、 w_1 和 w_2 之间道路互通,巡视器可以直接通过;目标 o_{bject0} 和

o_{bject1} 在4个地点方向均可见;初始状态下,着陆器L位于 w_0 ,巡视器R位于 w_3 ,任务目标是对 w_2 的土壤进行采样并将分析数据传回着陆器、对 w_3 的岩石采样并将分析数据传回着陆器、对 o_{bject1} 拍摄高分照片并将数据传回着陆器。利用Sapa规划器^[18]对该问题进行求解,得到的规划结果如图7中的下部分所示。由此,巡视器就可以在不同时刻执行对应的动作以完成任务目标。同时,根据定义2中的状态回退函数 $\Gamma(s_1, a)$,可以计算得到部分状态集合。当该规划执行失效时,触发RPDS进行规划修复。



- 1、0.0: (calibrate R camera objective1 w3) [5.0]
- 2、5.0: (take_image R w3 objective1 camera high_res) [7.0]
- 3、12.0: (communicate_image_data R L objective1 high_res w3 w0) [15.0]
- 4、27.0: (sample_rock R rstore w3) [8.0]
- 5、35.0: (navigate R w3 w1) [5.0]
- 6、35.0: (drop R rstore) [1.0]
- 7、40.0: (navigate R w1 w2) [5.0]
- 8、45.0: (communicate_rock_data R L w3 w2 w0) [10.0]
- 9、55.0: (sample_soil R rstore w2) [10.0]
- 10、65.0: (communicate_soil_data R L w2 w2 w0) [10.0]

图7 仿真场景示意及规划结果

Fig. 7 Simulation scenario and its planning results

考虑火星巡视器领域特点,其规划执行失败类型主要分为两种,一是固有条件的破坏,如 w_1 和 w_2 之间的道路被破坏,导致巡视器无法直接从 w_1 到达 w_2 ;另一种是可变条件的改变,如巡视器位置的变化、电能的存储变化等。当规划失效时,例如,若由于某种原因 w_3 和 o_{bject1} 之间不可互见,使得第1个动作(校准相机)的前提条件被破坏,从而使巡视器处于闲置状态。巡视器的感知状态与规划结果中的预估状态存在差异,RPDS就利用这些不同,有目的地向着对应的部分状态生成并扩展节点,并通过不断地搜索和冲突消解,完成规划修复问题的求解。

仿真中,通过在不同执行时刻手动植入意外事件来模拟规划执行失效,并采用RPDS进行修复。仿真测试结果如表2、图8和图9所示。其中,RP为文献[15]中的方法,在该仿真算例中,构造的修复结构树长度为3,深度为4。代码语言为Java,测试环境为I7-6700 CPU@ 3.40 GHz和8 GB内存。

从仿真结果可以看出，本文提出的基于状态差异的任务规划修复方法RPDS不仅能够处理固有条件被破坏、可变条件被改变的动作执行错误，而且相较于RP方法，既减少了因为构造修复结构树而增加的节点扩展数，且在大部分情况下缩短了修复时间，尤其是对于回退目标状态包含失败动作的前提条件情况，RPDS能够提升大约62%的效率。此外，与重规划方法Sapa的结果比较，RPDS提高了巡视器从执行失败中恢复的效率。

表 2 不同方法测试结果对比

Table 2 Comparison of test results of different methods

问题编号	修复耗时/ms	扩展节点数	修复动作数	规划稳定性
1	0.198/0.561	2/260	1/1	1/1
2	0.213/0.514	3/260	1/1	1/1
3	0.243/0.572	5/260	2/2	1/1
4	0.458/2.055	16/260	4/4	0.8/0.8
5	0.517/1.316	9/369	1/1	0.9/0.9
6	0.587/1.236	8/122	3/3	0.833/0.833
7	0.461/1.189	5/155	2/2	0.833/0.833
8	0.549/1.457	8/122	3/3	0.833/0.833
9	0.386/1.190	5/155	2/2	0.833/0.833
10	0.482/—	5/—	1/—	1/—
11	0.727/0.927	14/305	2/2	1/1
12	0.595/0.821	14/305	2/2	1/1
13	0.590/0.825	14/305	2/2	1/1
14	0.474/0.725	2/172	2/1	1/1

注：中间数据遵循RPDS/RP的形式，例如0.198/0.561表示RPDS耗时0.198 ms，RP耗时0.561 ms。

果可知，RPDS在快出给出修复解的同时，仍然能够以较少的动作保障规划稳定性。从第10例规划修复问题可以看出，相较于文献[15]中提出的规划修复算法RP，本文提出的方法RPDS能够处理更多的规划执行失败情况。可能的原因如下：RP在每次修复时，都是将已有规划结果按固定长度划分后，在限定的搜索空间内，通过后向搜索找到新的动作序列完成修复。那么对于每段中处于右端点的动作，在该限定的搜索空间内没有对应的子节点，从而导致RP无解的情况，或者该动作能够在下一段规划结果中找到规划解，但是由于搜索深度的限制，RP仍然无法给出对应的修复方案。而对于本文提出的RPDS方法来说，直接利用整个已有规划结果，从中提取部分状态，为规划修复提供修复目标，而不需要提前建立小范围内的搜索空间，从而避免了规划结果分段所导致的断层节点动作无法修复的问题。

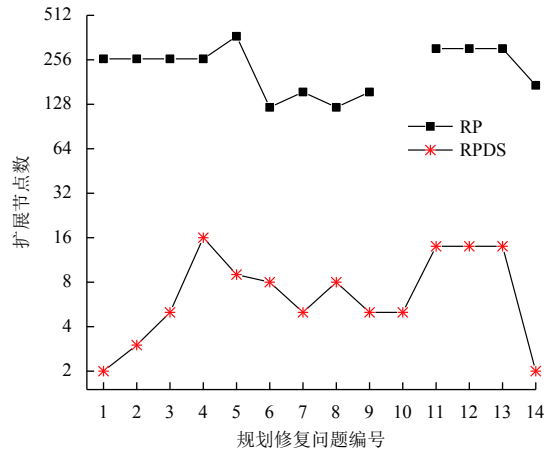


图 9 不同方法的扩展节点数对比

Fig. 9 Comparison of the number of expansion nodes in different methods

RPDS能够快速解决问题，主要是因为它能充分利用已有规划中的动作信息，构建部分状态，明确其与实际状态之间的差异，有目的地进行搜索，同时在已知回退目标状态的前提下，通过节点生成和冲突消解方法的控制，能够减少节点扩展数、缩减搜索空间的大小，从而提高修复效率。

5 结论

火星表面凹坑、凸起、坡度和石块的随机分布，土壤松软、厚度不均等复杂地貌，以及自身电子设备的意外故障等，均容易导致火星巡视器任务执行失败。针对该问题，本文提出了适用于火星巡视器的基于状态差异的快速任务规划修复方法RPDS，完成了以下工作：①利用火星巡视器的感知状态与动作执行的

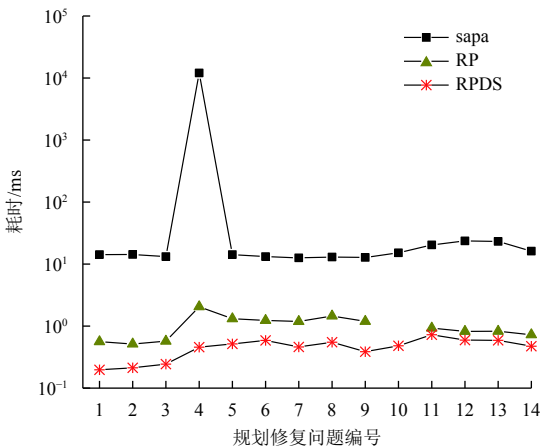


图 8 不同方法的运行时间对比

Fig. 8 Comparison of operation time of different methods

采用文献[11]中稳定性的评价方法，统计两个规划结果中互不相同的动作数之和，从表2的最后两列结

必要状态之间的差异,在已有规划的基础上,构建了不同时刻的部分状态,为规划修复提供可选修复目标;②基于部分状态,给出了火星巡视器快速规划修复策略;③利用部分状态和实际状态之间的差异,提出了基于状态差异的搜索空间删减方法,有目的地生成、扩展节点并消解冲突。通过仿真实验验证,RPDS能够快速有效地完成任务规划修复,并能够保障规划稳定性,可以为航天器自主快速应对执行时的突发事件提供技术支撑。

参 考 文 献

- [1] MAIMONE M W, LEGER P C, BIESIADECKI J J. Overview of the Mars exploration rovers' autonomous mobility and vision capabilities[C]//IEEE International Conference on Robotics and Automation. Rome, Italy: Space Robotics Workshop, 2007: 1-8.
- [2] 李群智,贾阳,彭松,等. 月面巡视探测器任务规划顶层设计与实现[J]. 深空探测学报(中英文), 2017, 4(1): 58-65.
LI Q Z, JIA Y, PENG S, et al. Top design and implementation of the lunar rover mission planning[J]. Journal of Deep Space Exploration, 2017, 4(1): 58-65.
- [3] PÉREZ-AYÚCAR M, ASHMAN M, ALMEIDA M, et al. The Rosetta science operations and planning implementation[J]. *Acta Astronautica*, 2018, 152: 163-174.
- [4] 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.
- [5] 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.
- [6] BRESINA J, DEARDEN R, MEULEAU N, et al. Planning under continuous time and resource uncertainty: a challenge for AI[C]//Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann Publishers Inc., 2002: 77-84.
- [7] 徐瑞,陈超,崔平远,等. 航天器自主任务规划修复技术研究进展[J]. 宇航学报, 2019, 40(7): 733-741.
XU R, CHEN C, CUI P Y, et al. Research on spacecraft autonomous mission plan repair[J]. *Journal of Astronautics*, 2019, 40(7): 733-741.
- [8] NEBEL B, KOEHLER J. Plan reuse versus plan generation: a theoretical and empirical analysis[J]. *Artificial Intelligence*, 1995, 76(1): 427-454.
- [9] CHIEN S, KNIGHT R, STECHERT A, et al. Using iterative repair to improve the responsiveness of planning and scheduling[C]//Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling. Menlo Park, California: The AIAA Press, 2000.
- [10] CHEN C, XU R, ZHU S Y, et al. RPRS: a reactive plan repair strategy for rapid response to plan failures of deep space missions[J]. *Acta Astronautica*, 2020, 175: 155-162.
- [11] FOX M, GEREVINI A, LONG D, et al. Plan Stability: replanning versus plan repair[C]//ICAPs'06: Proceedings of the Sixteenth International Conference on International Conference on Automated Planning and Scheduling. Menlo Park, California: The AAAI Press, 2005: 212-221.
- [12] GEREVINI A, SERINA I. Fast plan adaptation through planning graphs: local and systematic search techniques[C]//Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems. Breckenridge, CO: [s. n.], 2000.
- [13] SCALA E, MICALIZIO R, TORASSO P. ReCon: an online task reconfiguration approach for robust plan execution[C]//The Sixth International Conference on Agents and Artificial Intelligence (ICAART). ESEO, Angers, France: [s. n.], 2014.
- [14] GALLIEN M, INGRAND F, LEMAI S. Robot actions planning and execution control for autonomous exploration rovers[C]//International Workshop on Planning under Uncertainty for Autonomous Systems. Monterey, California: [s. n.], 2005.
- [15] GUZMAN C, CASTEJON P, ONAINDIA E, et al. Reactive execution for solving plan failures in planning control applications[J]. *Integrated Computer Aided Engineering*, 2015, 22(4): 343-360.
- [16] GHALLAB M, NAU D, TRAVERSO P. Automated planning: theory and practice[M]. Amsterdam, Boston: Elsevier/Morgan Kaufmann, 2004.
- [17] FOX M, LONG D. PDDL2.1: an extension to PDDL for expressing temporal planning domains[J]. *Journal of Artificial Intelligence Research*, 2003, 20(20): 61-124.
- [18] DO M B, KAMBHAMPATI S. Sapa: a multi-objective metric temporal planner[J]. *Journal of Artificial Intelligence Research*, 2003, 20(20): 155-194.

作者简介:

陈超(1994-),男,博士生,主要研究方向:航天器任务规划、航天器任务重规划。

通讯地址:北京市海淀区中关村南大街5号北京理工大学宇航学院22号信箱(100081)

E-mail: p_chenchao@126.com

徐瑞(1975-),男,教授,博士生导师,主要研究方向:航天器任务规划、自主导航、智能控制。**本文通讯作者。**

通讯地址:北京市海淀区中关村南大街5号北京理工大学宇航学院22号信箱(100081)

E-mail: xurui@bit.edu.cn

Fast Mission Plan Repair Method for Mars Rover Based on State Difference

CHEN Chao^{1,2}, XU Rui^{1,2}, LI Zhaoyu^{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: The uncertainty of Mars environment and the difficulty to predict the failure of electronic equipment will seriously affect the actual effect of the pre-designed plan of the rover on Martian surface, resulting in plan execution failure. To solve this problem, a fast mission plan repair method based on state difference is proposed based on the fact that there is long delay between the Mars rover and the ground station. This method constructs partial states at different times by extracting the key information from the existing plan, lying on the difference between the perception state and the necessary state of action execution. And then the rapid mission plan repair strategy for Mars rover based on the partial state is presented. In this strategy, to improve the efficiency of plan repair, the search space deletion method based on the state difference between the actual state and the partial state is proposed. Simulation results demonstrate that this method can not only improve the efficiency of mission plan repair, but also ensure the plan stability.

Keywords: plan repair; Mars rover; partial state; state difference

Highlights:

- Partial states are constructed by extracting the key information from the existing plan, lying on the difference between the perception state and the necessary state of action execution.
- A fast plan repair strategy of Mars rover is presented based on the partial state.
- A search guiding method is proposed, which can generate search nodes selectively according to the differences between partial state and real state.
- The rapidity of the method is evaluated, and it is found that the proposed plan maintains good plan stability.

[责任编辑：宋宏，英文审校：朱恬]