

**课程实验报告**

|  |  |
| --- | --- |
| 课程名称 | 自动规划 |
| 实验名称 | 数值-时间规划：移车问题 |

|  |  |
| --- | --- |
| 学 号： | 58122231 |
| 姓 名： | 陆文韬 |
| 学 院： | 人工智能学院 |
| 专 业： | 人工智能 |
| 指导教师： | 张志政 |
| 实验日期： | 2024-2025秋季学期 |

1. 规划问题定义
2. 流的描述：

本规划问题建模了一个复杂的车辆运输物流系统，包含多个相互关联的数据流。主要流包括物理流（车辆在道路网络中的移动、包裹的装载和卸载）、信息流（包裹目的地信息、道路状况信息、燃料水平监控）、时间流（动作持续时间、总体时间推进、延迟计算）以及资源流（燃料消耗、维护成本累积）。

物理流体现在车辆通过道路网络在不同地点间移动，同时承载包裹从起始位置运输到目标位置。信息流确保系统能够追踪每个包裹的当前位置和目标位置，监控道路状况以影响行驶策略，实时了解车辆燃料状态。时间流通过持续动作建模，每个动作都有明确的执行时间，系统能够计算包裹的交付延迟。资源流管理燃料的消耗和补充，以及基于道路状况和操作类型的维护成本累积。

1. 动作描述：

系统定义了四个核心的持续动作，每个动作都有明确的参数、持续时间、前置条件和效果。

**驾驶动作（drive）**：允许车辆沿指定道路从一个位置移动到另一个位置。动作的持续时间根据道路距离和固定速度（5单位/时间）计算。前置条件要求车辆必须在起始位置，道路必须连接两个位置，车辆必须有足够燃料完成行程。动作效果包括车辆位置的改变、燃料的消耗（按距离的0.2倍消耗）、行驶时间和总时间的增加，以及基于道路状况的维护成本累积（好路1.0单位，坏路3.0单位）。

**装载动作（load）**：实现车辆在特定位置装载包裹，固定持续时间1.5时间单位。前置条件确保车辆和包裹都在同一位置，且包裹尚未被装载。动作效果是包裹从位置移除并装载到车辆上，同时增加总时间和少量维护成本。

**卸载动作（unload）**：在包裹的目标位置卸载包裹，持续时间1.0时间单位。前置条件要求车辆已装载该包裹且位于包裹的目标位置。动作效果包括包裹从车辆移除并放置在位置上，计算并记录交付延迟，增加总时间和维护成本。

**加油动作（refuel）**：当车辆燃料低于20单位时进行加油，持续时间3.0时间单位。动作将燃料水平重置为100单位，恢复has-fuel状态，并增加相应的时间和维护成本。

1. 初始环境：

初始环境设定了一个包含5个地点（l1到l5）的交通网络，通过6条不同状况的道路连接。两辆车辆v1和v2分别位于l1和l5，初始燃料水平为80和60单位。

四个包裹的初始配置展现了问题的复杂性：p1位于l1需送达l3，p2需要在时间点5动态出现在l2并送达l4，p3位于l3需送达l5，p4需要在时间点10动态出现在l4并送达l1。这种设计包含了静态包裹（p1和p3立即可用）和动态包裹（p2和p4在特定时间出现），模拟了现实物流中的动态订单到达。

道路网络的设计考虑了不同的距离和状况：r1连接l1-l2（10单位，状况良好），r2连接l2-l3（15单位，状况不佳），r3连接l3-l4（8单位，状况良好），r4连接l4-l5（12单位，状况良好），r5提供l1-l5的直接连接（20单位，状况不佳），r6提供l2-l4的捷径（5单位，状况良好）。

所有数值型变量（时间、成本、延迟）都初始化为零，为动态计算提供基础。

1. 目标环境：

目标环境要求实现完整的物流任务：所有四个包裹必须送达各自的目标位置（p1到l3，p2到l4，p3到l5，p4到l1），同时两辆车辆都需要返回各自的基地位置（v1返回l1，v2返回l2）。

这个目标设计体现了现实物流操作的完整性要求，不仅要完成所有交付任务，还要确保资源（车辆）回到适当位置以便后续使用。

1. 规划生成机制（规划算法）介绍：

本问题采用了现代AI规划中的时序规划（Temporal Planning）方法，支持持续动作、数值流和时序约束。规划算法需要处理多个复杂特性：

**时序推理**：算法必须管理动作的持续时间和时序约束，确保动作的开始、执行过程和结束时刻的条件都得到满足。这要求算法维护一个时间线，跟踪所有实体在不同时间点的状态。

**数值规划**：系统中的燃料水平、距离、成本等数值变量需要专门的数值推理能力。算法必须能够处理数值计算、比较和优化。

**多目标优化**：目标函数是多个因素的加权组合，包括总时间、维护成本和交付延迟。规划算法需要找到在这个复合度量下的最优解。

**动态约束处理**：时序初始文字（timed initial literals）要求算法能够处理在特定时间点变为真的条件，这增加了规划搜索的复杂性。

典型的求解方法包括基于搜索的前向规划算法（如时序快速前向算法TFF）、基于编译的方法（将时序问题转换为经典规划问题）或基于约束满足的方法。

1. 规划程序设计与实现
2. 介绍
   1. 编程语言：

本规划问题使用PDDL（Planning Domain Definition Language）语言编写，这是人工智能规划领域的标准建模语言。PDDL提供了声明式的问题描述方式，允许规划师专注于问题的逻辑结构而不是具体的求解算法。

程序使用PDDL 2.1版本的扩展特性，包括类型系统（:typing）、数值流（:numeric-fluents）、持续动作（:durative-actions）和时序初始文字（:timed-initial-literals），这些特性使得建模更加精确和现实。

* 1. 运行环境：

程序需要在支持PDDL 2.1扩展特性的规划器中运行，推荐的环境包括：

OPTIC规划器：专门设计用于处理时序和数值规划问题

POPF规划器：支持部分观察和时序规划

TFD规划器：支持时序和数值流的规划器

VAL验证工具：用于验证规划解的正确性

运行环境需要支持Linux或Unix系统，具备足够的内存来处理搜索空间的探索。

1. 运行情况说明
   1. 输入输出说明：

**输入**：程序接受两个主要输入文件。领域文件（domain file）定义了车辆运输领域的类型、谓词、函数和动作，建立了问题的通用结构。问题文件（problem file）定义了具体的实例，包括对象、初始状态、目标状态和优化目标。

**输出**：规划器产生一个时序动作序列，每个动作包含动作名称、参数、开始时间和持续时间。典型的输出格式为：

0.00: (drive v1 r1 l1 l2) [2.00]

2.00: (load v1 p1 l2) [1.50]

3.50: (drive v1 r2 l2 l3) [3.00]

...

同时输出最终的目标函数值，显示总成本的组成部分。

* 1. 主要数据结构说明：

**状态表示**：系统状态由谓词集合和数值函数值组成。谓词包括车辆位置、包裹位置、装载关系等布尔关系。数值函数包括燃料水平、时间、成本等连续值。

**动作表示**：每个动作由参数化的模板定义，包括参数类型、持续时间表达式、前置条件和效果。动作实例化时绑定具体的对象参数。

**搜索空间**：搜索节点包含当前状态、已执行动作序列、当前时间和累积成本。搜索树的分支对应可执行动作的选择。

**约束网络**：时序约束通过时间点网络（STN）管理，确保所有时序约束得到满足。

* 1. 运行案例：

考虑一个简化的执行序列：规划器首先分析初始状态，发现v1在l1可以立即装载p1，而v2在l5可以装载p3。由于p2和p4有时序约束，规划器需要安排车辆在适当时间到达相应位置。

一个可能的优化解决方案是：v1装载p1后通过r1-r2路径送达l3（考虑r2的不良状况），然后前往l2等待p2的到达。v2装载p3直接送达l5，然后前往l4等待p4。整个过程需要仔细平衡时间成本、燃料消耗和维护成本。

1. 分析
2. 通用性说明（从算法和规划语言两方面说明）：

**算法通用性**：时序数值规划算法具有很强的通用性，能够处理广泛的实际应用问题。算法的核心搜索和约束处理机制可以适用于制造调度、资源分配、机器人任务规划等多个领域。数值推理能力使算法能够处理涉及连续资源的问题，而时序推理能力适用于所有涉及时间因素的规划问题。

**规划语言通用性**：PDDL作为标准化的规划语言，提供了强大的表达能力和良好的可移植性。本问题使用的建模技术（类型系统、数值流、持续动作）可以直接应用于其他领域的问题建模。领域独立的特性使得相同的求解器可以处理不同应用领域的问题。

1. 完备性说明：

理论上，基于系统搜索的时序规划算法是完备的，即如果问题有解，算法能够找到解。本问题的完备性依赖于几个因素：搜索空间的有限性（通过适当的状态空间剪枝确保），启发式函数的可采纳性（确保不会错过最优解），以及约束传播的完整性（确保所有时序和数值约束都被正确处理）。

然而，实际实现中可能因为计算资源限制而无法达到理论完备性。内存限制可能导致搜索过程中状态空间的不完全探索，时间限制可能阻止算法找到存在的解。

1. 计算复杂度说明：

时序数值规划问题属于PSPACE-complete复杂度类，比经典规划的NP-complete更加困难。复杂度的主要来源包括：

**状态空间大小**：随着对象数量指数增长，本问题中5个位置、2辆车、4个包裹和6条道路产生的状态组合数量巨大。

**时序约束处理**：维护时序一致性需要额外的计算开销，特别是在处理动作的开始、执行和结束条件时。

**数值计算**：燃料水平、成本函数等数值变量的处理增加了计算复杂度，特别是在需要精确数值比较时。

**优化目标**：多目标优化要求算法探索更大的搜索空间以找到帕累托最优解。

实际运行时间随问题规模超线性增长，大规模实例可能需要启发式搜索和近似算法来在合理时间内获得解。

1. 正确性说明：

规划解的正确性通过多个层面保证：

**语法正确性**：PDDL语法检查确保领域和问题定义的语法正确性，类型系统防止类型错误。

**语义正确性**：每个动作的前置条件和效果都经过仔细设计，确保状态转换的逻辑正确性。数值计算（如燃料消耗、成本累积）基于现实物理约束。

**约束满足**：解必须满足所有时序约束、数值约束和逻辑约束。VAL等验证工具可以独立验证解的正确性。

**目标达成**：解必须使所有目标条件在最终状态下为真，包括包裹交付和车辆返回基地。

1. 最优性：

本问题采用多目标优化，最优性通过加权目标函数定义。当前的权重设置（总时间权重2，维护成本权重1，延迟权重0.5）体现了对时间效率的重视。

**帕累托最优性**：真正的最优解应该考虑所有目标之间的权衡，找到帕累托最优解集。当前的加权方法可能错过某些在特定权重下更优的解。

**算法最优性**：A\*等最优搜索算法可以保证找到最优解，但需要可采纳的启发式函数。实际中常用的启发式可能不满足可采纳性，导致次优解。

**近似最优性**：对于大规模问题，可能需要接受在合理时间内找到的近似最优解，通过设置适当的搜索界限和启发式策略。

1. 不足说明：

当前模型和实现存在几个重要不足：

**建模简化**：现实物流系统的许多复杂因素被简化或忽略，如交通拥堵、天气影响、车辆容量限制、驾驶员工作时间限制等。道路的双向性质没有明确建模，可能导致不现实的路径选择。

**可扩展性限制**：随着车辆、包裹和位置数量的增加，搜索空间呈指数增长，算法性能急剧下降。当前的建模方法难以处理大规模的实际物流网络。

**动态性处理不足**：虽然包含了时序初始文字，但对于更复杂的动态环境（如动态交通状况、紧急事件）的处理能力有限。

**不确定性缺失**：现实中的燃料消耗、行驶时间等都存在不确定性，但当前模型假设了确定性环境，可能导致实际执行时的问题。

**人机交互限制**：系统缺乏与人类操作员的有效交互机制，无法处理执行过程中的异常情况或动态调整需求。

**求解器依赖**：问题的解决完全依赖于外部规划器的性能，不同求解器可能产生差异很大的结果，且大多数高性能求解器不是开源的。

尽管存在这些不足，本规划问题仍然很好地展示了现代AI规划技术在复杂物流问题中的应用潜力，为进一步的研究和实际应用提供了坚实的基础。