

**课程实验报告**

|  |  |
| --- | --- |
| 课程名称 | 自动规划 |
| 实验名称 | 车辆调度规划实验 |

|  |  |
| --- | --- |
| 学 号： | 58122231 |
| 姓 名： | 陆文韬 |
| 学 院： | 人工智能学院 |
| 专 业： | 人工智能专业 |
| 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：

本实验实现了一个基于PDDL的车辆调度规划问题。该问题模拟了多层停车场环境，其中车辆需要在不同的车库（层）之间进行调度，以达到指定的目标配置。这是一个典型的状态空间搜索问题，体现了人工智能规划中的核心概念，包括状态表示、动作定义、前置条件和效果描述。

问题的核心挑战在于车辆移动受到物理约束的限制。每个车库采用类似栈的结构，车辆只能从顶部位置进行移动，这反映了现实停车场的物理特性。车辆可以在同一车库内的不同位置间移动，也可以跨车库转移，但都必须遵守位置邻接关系和占用状态的约束。

1. 动作描述：

本规划问题定义了两种基本动作类型，每种动作都有明确的前置条件和效果，体现了PDDL语言的声明式特性：

**车库间转移动作（transfer\_between\_garages）**：此动作允许车辆在不同车库之间移动，但仅限于顶部位置。动作的前置条件确保只有位于顶部槽位的车辆才能进行跨车库转移，这模拟了实际停车场中车辆必须从最上层移出的物理约束。动作的效果包括更新车辆的位置信息和相应槽位的占用状态。

**车库内移位动作（shift\_within\_garage）**：此动作允许车辆在同一车库内的相邻位置间移动。这种设计考虑了车辆调整位置的需求，例如为其他车辆让出通道或优化停车空间利用。动作使用析取前置条件来处理双向移动的可能性，体现了PDDL在表达复杂约束关系方面的能力。

1. 初始环境：

实验设定的初始环境包含四辆车辆（v1、v2、v3、v4）分布在三个车库（g1、g2、g3）中。具体配置为：车辆v1位于车库g1的顶部槽位s2，车辆v2和v3分别位于车库g2的底部槽位s1和顶部槽位s2，车辆v4位于车库g3的底部槽位s1。

这种初始配置具有一定的复杂性，车库g2同时容纳两辆车辆，形成了一个需要仔细规划的移动序列。槽位s1和s2的邻接关系定义了车辆在垂直方向上的移动可能性，而s2被标记为顶部槽位，限制了某些移动操作的执行条件。

1. 目标环境：

目标状态要求对车辆进行重新分配：车辆v1需要移动到车库g2的顶部槽位，车辆v2移动到车库g1的底部槽位，车辆v3转移到车库g3的顶部槽位，车辆v4则需要移动到车库g1的顶部槽位。

这个目标配置需要协调多个车辆的移动，特别是车库g1最终需要容纳两辆车辆（v2在底部，v4在顶部），这要求规划算法能够找到合适的中间状态和移动序列。目标状态的复杂性测试了规划系统处理多约束优化问题的能力。

1. 规划生成机制（规划算法）介绍：

PDDL作为标准的规划领域定义语言，支持多种先进的规划算法。典型的规划器如Fast Forward（FF）、LAMA、或BFWS等可以处理这类问题。这些规划器通常采用启发式搜索算法，结合前向状态空间搜索和智能启发函数来高效地探索解空间。

规划过程从初始状态开始，通过应用可执行的动作生成后继状态，直到达到满足目标条件的状态。算法使用启发函数估计从当前状态到目标状态的距离，指导搜索朝着最有希望的方向进行。对于本问题，规划器需要处理动作之间的依赖关系，例如某些车辆必须先移动才能为其他车辆腾出空间。

1. 规划程序设计与实现
2. 介绍
   1. 编程语言：PDDL（规划领域定义语言）。
   2. 运行环境：实验可以在多种PDDL规划器上运行，包括Fast Forward、LAMA、BFWS等主流规划器。
3. 运行情况说明
   1. 输入输出说明：规划器的输入包含两个PDDL文件：域文件（domain file）定义了问题的通用结构，包括对象类型、谓词和动作；问题文件（problem file）定义了具体的实例，包括对象、初始状态和目标状态。规划器的输出是一个动作序列，详细描述了从初始状态到目标状态所需的每个步骤。
   2. 主要数据结构说明：PDDL使用基于逻辑的数据结构。对象（objects）表示问题中的实体，如车辆、车库和槽位。谓词（predicates）表示对象之间的关系和属性，如位置关系和占用状态。动作（actions）定义了状态转换规则，包含参数、前置条件和效果。这种结构化的表示方式使得问题描述清晰且易于验证。
   3. 运行案例：规划器可能生成如下动作序列解决此问题：首先执行shift\_within\_garage动作将v3从g2的s2移动到s1，然后执行transfer\_between\_garages动作将v1从g1转移到g2，接着进行一系列类似的移动操作，最终达到目标配置。具体的动作序列取决于规划器采用的搜索策略和启发函数。
4. 分析
5. 通用性说明（从算法和规划语言两方面说明）：

**算法通用性**：基于PDDL的规划方法具有很强的通用性。同样的算法框架可以应用于各种不同的规划问题，从物流调度到机器人路径规划，再到资源分配优化。规划算法的核心思想—状态空间搜索配合启发函数—是领域无关的，这使得开发通用规划器成为可能。

**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。它支持类型系统、否定前置条件、析取条件等高级特性，能够描述现实世界中的复杂约束关系。PDDL的标准化特性意味着同一个问题描述可以在不同的规划器上运行，促进了算法比较和性能评估。

1. 完备性说明：

基于PDDL的规划方法在理论上具有完备性保证。如果问题存在解，现代规划器如BFWS或LAMA最终能够找到解。这种完备性基于状态空间搜索的完备性理论：在有限状态空间中，系统化的搜索算法能够遍历所有可达状态。

对于本车辆调度问题，状态空间是有限的，因为车辆数量、车库数量和位置数量都是固定的。每个状态可以用车辆位置的组合来唯一标识，而合法位置组合的数量是有限的。因此，如果目标状态从初始状态可达，规划器必定能够找到相应的动作序列。

1. 计算复杂度说明：

车辆调度问题的计算复杂度主要取决于状态空间的大小和分支因子。设有n辆车辆、m个车库、每个车库有k个槽位，则状态空间的上界为(m×k)^n。然而，实际的状态空间会因为物理约束（如槽位占用唯一性）而显著减小。

分支因子取决于每个状态下可执行的动作数量。在最坏情况下，每辆位于顶部槽位的车辆都可以移动到其他车库或同一车库的其他位置，导致较高的分支因子。现代启发式规划算法通过智能的搜索策略和剪枝技术，能够在合理时间内求解中等规模的此类问题。

1. 正确性说明：

PDDL规划方法的正确性有多层保障。首先，PDDL的声明式语义确保了问题描述的精确性：每个动作的前置条件和效果都有明确的逻辑含义。其次，规划器的搜索算法只会应用满足前置条件的动作，确保每个状态转换都是合法的。

对于本实验，正确性可以通过验证生成的动作序列来确认：检查每个动作的执行是否满足其前置条件，以及最终状态是否符合目标规范。PDDL的形式化特性使得这种验证过程可以自动化进行，增强了解的可靠性。

1. 最优性：

规划解的最优性取决于优化目标的定义。如果以动作数量最少为目标，某些规划器（如使用A\*搜索的规划器）能够保证找到最优解。然而，大多数现代规划器为了提高效率，采用满意化搜索（satisficing search），优先寻找可行解而非最优解。

对于车辆调度问题，最优性的定义可能包括最少移动次数、最短执行时间或最小资源消耗。本实验的PDDL描述可以通过修改动作代价或添加度量函数来支持不同的优化目标，但这需要使用支持度量规划的高级PDDL功能。

1. 不足说明：

当前的PDDL实现存在几个局限性。首先，问题规模的扩展性有限：当车辆数量和车库数量大幅增加时，搜索空间呈指数增长，可能导致求解时间过长。其次，模型的现实性约束较为简化：实际停车场可能涉及车辆尺寸差异、通道宽度限制、时间窗口约束等复杂因素，这些在当前模型中未得到充分体现。

此外，PDDL作为静态规划语言，难以处理动态环境变化，如新车辆的实时到达或突发的设备故障。对于这类动态规划问题，可能需要结合在线规划或重规划技术。最后，当前的动作模型假设所有操作都能成功执行，没有考虑执行失败或不确定性因素，这限制了其在实际应用中的鲁棒性。