

**课程实验报告**

|  |  |
| --- | --- |
| 课程名称 | 自动规划 |
| 实验名称 | 时序电梯规划 |

|  |  |
| --- | --- |
| 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：

本程序实现了一个基于时间的电梯调度规划问题，这是对经典电梯问题的重要扩展。与传统的离散规划不同，此问题引入了时间维度，使得问题更加贴近现实世界的电梯运行场景。

问题的核心在于协调多部电梯的运行，将分布在不同楼层的多名乘客运送到目标楼层，同时优化总执行时间。这个问题属于多智能体调度问题的范畴，需要考虑资源竞争（电梯容量）、时间约束（不同速度的移动时间）以及优化目标（最小化总时间）。

与静态规划问题相比，时间规划问题的状态不仅包含位置信息，还包含时间信息。每个动作都有明确的持续时间，这使得问题的求解需要考虑动作的并行性和时序约束。例如，两部电梯可以同时运行，而乘客的上下电梯动作必须在电梯到达相应楼层后才能执行。

1. 动作描述：

系统定义了四个核心的持续时间动作，每个动作都具有明确的时间语义：

**电梯上行动作（move-up）**：控制电梯从当前楼层向上移动到相邻楼层。动作的持续时间由楼层间距离除以电梯速度计算得出，这完美地模拟了现实中电梯运行的物理约束。前置条件要求电梯必须在起始楼层，且目标楼层必须是相邻的上层楼层。效果是在动作开始时电梯离开当前楼层，在动作结束时到达目标楼层。

**电梯下行动作（move-down）**：与上行动作对称，控制电梯向下移动。关键区别在于楼层关系的判断，使用next谓词的不同参数顺序来确保移动方向的正确性。

**乘客登梯动作（board）**：处理乘客进入电梯的过程。动作持续时间由乘客的移动速度决定，反映了不同乘客可能具有不同的行动能力。前置条件要求电梯必须在整个动作期间都停留在相应楼层，这确保了乘客能够安全登梯。

**乘客离梯动作（leave）**：处理乘客离开电梯的过程。与登梯动作类似，但效果相反，将乘客从电梯内状态转换为在楼层的状态。

这四个动作的设计体现了时间规划的精髓：每个动作都有明确的开始和结束时刻，以及在整个执行期间必须满足的条件。

1. 初始环境：

初始状态描绘了一个五层建筑的电梯系统。建筑包含从n1到n5的五个楼层，楼层间通过next谓词建立相邻关系，形成了线性的楼层结构。

系统配置了两部电梯：电梯e1初始位置在一楼（n1），电梯e2初始位置在五楼（n5）。这种配置提供了良好的覆盖范围，两部电梯分别位于建筑的两端。

三名乘客的初始分布体现了典型的电梯使用场景：乘客p1和p2都在二楼（n2），而乘客p3在四楼（n4）。这种分布要求电梯系统进行复杂的调度决策。

时间参数的设定增加了问题的真实性。每名乘客具有不同的移动速度：p1需要2秒上下电梯，p2需要3秒，p3需要2秒。两部电梯也有不同的运行速度：e1的速度为2米/秒，e2的速度为3米/秒，这意味着e2是更快的电梯。

楼层间的距离设定反映了建筑的物理结构：n1到n2之间3米，n2到n3之间4米，n3到n4之间4米，n4到n5之间3米。注意程序同时定义了上行和下行的距离，确保了双向移动的准确计算。

1. 目标环境：

目标状态要求所有三名乘客都到达一楼（n1）。这是一个典型的疏散场景，可能模拟紧急情况下的人员撤离或者下班时段的集中下楼需求。

目标的设定看似简单，但实际执行需要复杂的协调。由于乘客分布在不同楼层，电梯需要分别前往这些楼层接载乘客，然后统一运送到一楼。这个过程涉及多个决策点：使用哪部电梯接载哪些乘客，如何安排接载顺序以最小化总时间。

优化目标设定为最小化总执行时间，这使得问题从满足性规划转变为优化规划。系统不仅要找到可行解，还要在所有可行解中找到时间最优的解决方案。

1. 规划生成机制（规划算法）介绍：

本程序采用时间规划（Temporal Planning）算法，这是经典STRIPS规划的重要扩展。时间规划算法需要处理动作的持续时间、并行执行和时序约束等复杂因素。

算法的核心是时间状态空间搜索，其中每个状态不仅包含世界的逻辑状态，还包含当前时间信息。搜索过程中，算法需要维护一个时间线，跟踪各个动作的开始和结束时间，确保时序约束的满足。

时间规划算法通常采用前向搜索策略，从初始状态开始，逐步应用动作直到达到目标状态。关键挑战在于处理动作的并行性：多个动作可能同时执行，只要它们的前置条件和效果不冲突。这要求算法具备冲突检测和解决机制。

为了优化总执行时间，算法可能采用启发式搜索方法，如时间A\*算法。启发式函数需要估计从当前状态到目标状态的最小时间，这通常比经典规划的启发式函数更复杂，因为需要考虑时间因素。

1. 规划程序设计与实现
2. 介绍
   1. 编程语言：

程序使用PDDL 2.1标准编写，这是PDDL语言的重要扩展版本，引入了对时间和数值的支持。相比基础PDDL，PDDL 2.1增加了三个关键特性：typing（类型系统）、fluents（数值函数）和durative-actions（持续时间动作）。

类型系统通过:typing关键字启用，允许对象进行分类，提高了模型的表达能力和求解效率。数值函数通过:fluents关键字启用，支持数值计算和比较。持续时间动作通过:durative-actions关键字启用，这是时间规划的核心特性。

程序的语法结构遵循PDDL 2.1的标准格式，使用S-表达式（类似Lisp的语法）来定义领域和问题。这种结构化的表示方法使得程序既具有人类可读性，又便于计算机解析。

* 1. 运行环境：

程序需要在支持PDDL 2.1标准的时间规划求解器上运行。主要的求解器包括POPF（Partial Order Planning with Preferences and Fluents）、OPTIC（Optimising Preferences and Time-dependent Costs）、TFD（Temporal Fast Downward）等。

这些求解器通常运行在Linux或Unix环境下，通过命令行接口调用。求解器需要同时读取领域文件和问题文件，然后输出包含时间信息的解决方案。一些现代求解器还支持并行计算和分布式求解，以处理更大规模的问题。

1. 运行情况说明
   1. 输入输出说明：

**输入格式**：程序接受两个主要输入文件。领域文件定义了问题的通用结构，包括对象类型、谓词、函数和动作模式。问题文件定义了具体的实例，包括对象声明、初始状态、目标状态和优化指标。

与经典PDDL不同，PDDL 2.1的输入还包含数值信息，如速度、距离和时间等函数值的初始设定。这些数值参数直接影响动作的持续时间计算。

**输出格式**：求解器输出一个带时间戳的动作序列，称为时间规划图（Temporal Plan）。输出格式通常包含每个动作的开始时间、结束时间、动作名称和参数。例如：

0.000: (move-up e1 n1 n2) [1.500]

0.000: (move-down e2 n5 n4) [1.000]

1.500: (board p1 n2 e1) [2.000]

这表示两个移动动作在时刻0同时开始，登梯动作在时刻1.5开始。

* 1. 主要数据结构说明：

**时间状态表示**：每个状态包含两个主要组成部分：逻辑状态（谓词的真值分配）和时间状态（当前时间和正在执行的动作）。时间状态的维护是时间规划的关键挑战，需要跟踪所有活跃动作的进度。

**动作调度表**：系统维护一个动作调度表，记录所有已调度动作的开始和结束时间。这个数据结构用于检测动作间的时间冲突和资源竞争。

**约束网络**：为了处理复杂的时序约束，系统可能构建时间约束网络（Temporal Constraint Network），其中节点表示时间点，边表示时间间隔约束。

* 1. 运行案例：

一个典型的解决方案可能按以下时序执行：

**时刻0.000**：两部电梯同时开始移动。电梯e1从n1向上移动到n2（用时1.5秒，因为距离3米，速度2米/秒），电梯e2从n5向下移动到n4（用时1.0秒，因为距离3米，速度3米/秒）。

**时刻1.000**：电梯e2到达n4，开始接载乘客p3（用时2秒）。

**时刻1.500**：电梯e1到达n2，开始接载乘客p1（用时2秒）。

**时刻3.000**：乘客p3完成登梯，电梯e2开始向下移动到n1。

**时刻3.500**：乘客p1完成登梯，乘客p2开始登梯（用时3秒）。

这个案例展示了时间规划的并行性和精确的时序控制。

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**：时间规划算法具有很强的通用性，能够处理各种涉及时间和资源约束的问题。算法框架可以扩展到其他领域，如生产调度、机器人路径规划、项目管理等。通过修改领域定义，可以适应不同的应用场景。

该算法的通用性还体现在对不同规模问题的适应能力。通过增加电梯数量、楼层数量或乘客数量，可以测试算法在更复杂场景下的性能。数值参数的灵活配置使得模型能够准确反映不同的物理环境。

**规划语言通用性**：PDDL 2.1作为国际标准，保证了程序的可移植性和互操作性。程序可以在不同的求解器上运行，无需修改代码。同时，PDDL 2.1的表达能力足够强大，能够处理复杂的时间约束、数值计算和优化目标。

语言的模块化设计使得领域知识和问题实例可以分离，便于知识重用和问题变体的生成。这种分离还有利于领域专家和规划专家的协作。

1. 完备性说明：

时间规划算法在理论上具有完备性，即如果存在解决方案，算法一定能找到。这个保证基于几个关键因素：

**状态空间的有限性**：虽然引入了时间维度，但在有限的离散化精度下，状态空间仍然是有限的。每个逻辑状态与时间的组合构成一个独特的时间状态。

**搜索的系统性**：前向搜索算法系统地探索所有可能的动作序列，不会遗漏任何可达状态。时间约束的加入实际上减少了搜索空间，因为许多逻辑上可能的动作序列在时间上是不可行的。

**终止条件的明确性**：算法具有明确的终止条件（到达目标状态）和失败检测机制（搜索空间耗尽）。时间约束还提供了额外的剪枝机会，当部分计划的执行时间已经超过已知最优解时，可以提前终止该分支。

然而，完备性的实现依赖于合理的时间离散化和有限的时间范围。在实际应用中，可能需要设置时间上界以保证算法的终止。

1. 计算复杂度说明：

时间规划的计算复杂度明显高于经典规划，主要原因包括：

**时间复杂度**：最坏情况下的时间复杂度为O(b^d × T)，其中b是分支因子，d是解的深度，T是时间范围。相比经典规划的O(b^d)，增加了时间维度T。在本问题中，分支因子约为8-12（考虑所有可能的电梯和乘客动作），解的深度约为10-15步。

**空间复杂度**：需要存储时间状态空间，复杂度为O(|S| × T)，其中|S|是逻辑状态数量。时间约束网络的维护也需要额外的空间开销。

**数值计算开销**：每个动作的持续时间计算涉及浮点运算，比经典规划的布尔运算更耗时。时间约束的检查和传播也增加了计算负担。

实际性能可能因启发式函数和剪枝策略而显著改善。现代求解器采用的技术包括局部搜索、约束传播、并行计算等。

1. 正确性说明：

**语义正确性**：动作的前置条件和效果准确地模拟了电梯系统的行为。时间语义的使用确保了动作的执行遵循物理定律，如电梯不能瞬间移动，乘客不能在电梯运行时上下。

**约束一致性**：程序中的所有约束都是一致的，不存在矛盾的要求。楼层的相邻关系、距离设定、速度参数等都符合常识和物理规律。

**时序正确性**：动作的时间约束确保了执行顺序的合理性。例如，乘客必须在电梯到达后才能登梯，电梯必须在移动完成后才能在新楼层接载乘客。

**数值精度**：程序使用精确的数值计算来确定动作持续时间，避免了舍入误差可能导致的时序问题。

1. 最优性：

本程序明确追求时间最优性，通过:metric minimize (total-time)指标定义。这使得问题从满足性规划转变为优化规划，具有更高的求解要求。

**最优性保证**：采用最优搜索算法（如A\*）时，程序能够保证找到时间最优解。启发式函数需要满足可采纳性条件，即不高估实际代价。

**多目标考虑**：除了总时间外，还可以考虑其他优化目标，如能耗、乘客等待时间、电梯磨损等。多目标优化需要更复杂的算法框架。

**近似最优性**：在大规模问题中，为了平衡求解时间和解的质量，可能采用近似算法，在合理时间内找到接近最优的解。

1. 不足说明：

程序存在几个主要局限性，指出了未来改进的方向：

**建模简化**：当前模型没有考虑电梯容量限制、加速减速过程、开关门时间、电梯调度策略等现实因素。这些简化虽然使问题更易处理，但降低了模型的真实性。

**扩展性限制**：随着电梯数量、楼层数量和乘客数量的增加，问题的复杂度急剧上升。当前的精确算法可能无法处理大规模的实际应用。

**动态性缺失**：程序假设所有信息在规划开始时都是已知的，无法处理动态到达的乘客请求或设备故障等突发情况。现实的电梯系统需要在线重新规划能力。

**启发式函数**：缺乏专门针对电梯调度问题的高效启发式函数，可能导致搜索效率低下。开发领域特定的启发式知识是提高性能的关键。

**不确定性处理**：程序没有考虑执行时间的不确定性、设备故障的可能性等随机因素。在实际应用中，鲁棒性规划是重要需求。

这些不足为未来的研究和开发提供了明确的方向，包括更精确的建模、更高效的算法、更强的适应性和更好的用户接口。