

**课程实验报告**

|  |  |
| --- | --- |
| 课程名称 | 自动规划 |
| 实验名称 | 车辆搬运规划算法设计 |

|  |  |
| --- | --- |
| 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：该规划问题涉及将不同车库中的车辆根据目标配置转移至新的车库。每辆车的转移遵循车库容量的限制
3. 动作描述：每次行动涉及将一辆车从一个车库移动到另一个车库，且移动过程遵守车库最大容量的约束，并且对于每个车库而言，车辆只能从车库的一侧移动
4. 初始环境：初始环境中，车库中的车辆配置由用户输入，包括每个车库内的车辆列表。例如，车库1可能有两辆车，车库2有一辆车，车库3为空
5. 目标环境：目标环境是将车库中的车辆按指定目标配置排列，每个车库的车辆数量不能超过最大容量
6. 规划生成机制（规划算法）介绍：本实验使用广度优先搜索（BFS）算法进行规划。BFS从初始状态开始，探索所有可能的车辆移动，直到找到一个符合目标车库配置的方案
7. 规划程序设计与实现
8. 介绍
   1. 编程语言：Python
   2. 运行环境：程序在Windows 10系统下运行，使用Python 3.8版本，Tkinter用于图形界面显示，PIL用于加载和展示车库及汽车图像
9. 运行情况说明
   1. 输入输出说明：输入为初始和目标车库配置的车牌号以及每个车库的最大容量。输出为一个动作序列，表示每次移动的车辆和源车库、目标车库的索引
   2. 主要数据结构说明：车库的状态通过包含3个车库（每个车库用元组表示）的元组进行存储。BFS使用队列来保存待处理的状态和路径
   3. 运行案例：例如，初始车库配置为[["01", "02"], ["04","03"], []]，目标配置为[["04"], ["02", "03"], ["01"]]，程序会找到移动步骤，展示车辆从车库1和车库2之间的移动
10. 分析
11. 通用性说明（从算法和规划语言两方面说明）：该算法适用于大多数具有类似车库配置的状态转移问题，可以自定义车库的最大容量
12. 完备性说明：BFS算法是完备的，因为它保证会遍历所有可能的状态，直到找到目标状态为止，若无解则返回无解
13. 计算复杂度说明：BFS的计算复杂度为O(b^d)，其中b是每个车库中可能的移动数，d是从初始状态到目标状态所需的步数
14. 正确性说明：算法通过验证每次车辆转移是否符合车库容量约束来确保正确性，且最终能够达到目标状态
15. 最优性：BFS算法保证最优性，因为它总是首先探索路径最短的状态，确保找到最少移动次数的解决方案
16. 不足说明：该算法的局限性在于其对状态空间的搜索可能非常耗时，尤其是在车库数量和车辆数量较多时。可以考虑引入启发式搜索算法如A\*来优化性能。此外，该算法设定车库数量为3，如果问题要求别的车库数量，则无法解决问题。