UNIVERSITÀ DEGLI STUDI DI GENOVA

DIBRIS

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY,
BIOENGINEERING, ROBOTICS AND SYSTEM ENGINEERING

ARTIFICIAL INTELLIGENCE FOR ROBOTICS 2

First Assignment

AI Planning

Authors:

Borelli Simone - s4662264
Carlini Massimo - s4678445
Forni Matteo - s4673648
Gavagna Veronica - s5487110

Professors:

Mauro Vallati

May 4, 2023

Contents

1 Introduction

The following report for the AIRO2 course will describe how we have efficiently modelled, as much as possible, a robotic coffee shop solution for managing the orders made by the customers. In order to implement the AI plan regarding this scenario, we took advantage of the PDDL2.1 planning language features. Thanks to actions and durative-actions, we achieved every requirement the shop owner asked for, including 3 out of 4 optional extensions.
Unfortunately, we were not able to implement the last optional feature, due to a limitation of the chosen planner. This problem is discussed in the last section of the report.

2 Assignment Scenario

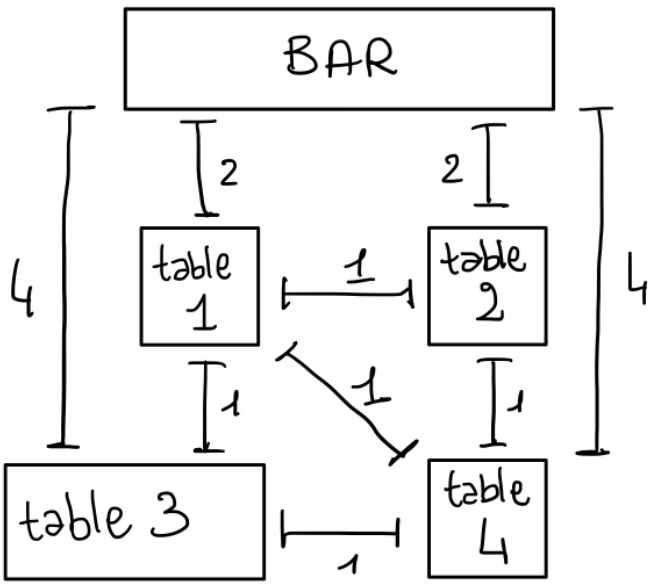


Figure 1: assignment scenario: each distance is expressed in meters

The coffee shop scenario is shown in figure 1 and the agents are:

- **One Barista**:
  it is in charge of preparing drinks. Cold drinks are prepared faster than hot ones. It assumes that all orders are known, so waiters do not need to take customer orders;
- **Two Waiters**:
  it is in charge of serving customers and cleaning tables when customers have already left the shop. The waiter robot can decide to grasp a drink with its gripper or to use a tray for carrying more drinks at once. When it uses the tray, the robot moves slower to improve balance. It is important to point out that the two waiters must not be at the same place for the same instant of time.

3 Installing and Running

The project works with LPG-TD or LPG++ planners, which supports Classical planning with durative-actions.
The planner reads as inputs a PDDL problem and domain files, and it provides a plan (a sequence of actions and durative actions, that allow the achievement of a goal). The plan is a time-stamped plan, so associated with each action, there will be the time at which that instance is executed.

notes:

- LPG is a planner based on local search and planning graphs that handles PDDL2.2 domains involving numerical quantities and durations.
- The planner does not support events, processes and concurrency. Also, its next version, LPG++, does not manage concurrency between actions and durative.

3.1 Installing

For installing the LPG planners, it is just necessary to download the folder from the official website (for LPG-TD `https://lpg.unibs.it/lpg/` and for LPG++ `http://helios.hud.ac.uk/scommv/storage/lpg++`) and place the problem and domain files in the same folder.

notes:

- Inside our GitHub repository (`https://github.com/Matteoforni1/AIRO2_Assignment1.git`), more detailed instructions are reported in the README file.

3.2 Running

The planning can be executed from the folder containing the planner, the problem and the domain files using the following command:

- If you choose LPG-TD planner:

$$./lpg - td\ -o\ < domainfile > .pddl\ -f\ < problemfile > .pddl\ -n\ 1$$

- If you choose LPG++ planner:

$$./lpg++\ -o\ < domainfile > .pddl\ -f\ < problemfile > .pddl\ -n\ 1$$

notes:

- There are several options for running the planner; the ones reported above are exactly the same as we have used for generating the 4 problem files in the zip file; other options are discussed in the **Performance** section.

4 Description of the problem

In the beginning, all the orders and the place, in which customers are sitting, are known. Then the barista starts making the drinks and once ready, the drinks are put on the bar where the waiter can pick them up and serve them to the customers. Important: the barista makes one drink, regardless of the type, per time. Finally, the waiter has to clean the table when the customers have already consumed their drinks. In addition, we briefly exposed the four problems.

4.1 Problem 1

There are 2 customers at table 2: they ordered 2 cold drinks. Tables 3 and 4 need to be cleaned.

4.2 Problem 2

There are 4 customers at table 3: they ordered 2 cold drinks and 2 warm drinks. Table 1 needs to be cleaned.

4.3 Problem 3

There are 2 customers at table 4: they ordered 2 warm drinks. There are also 2 customers at table 1: they ordered 2 warm drinks. Table 3 needs to be cleaned.

4.4 problem 4

There are 2 customers at table 4 and 2 customers at table 1: they all ordered cold drinks. There are also 4 customers at table 3: they all ordered warm drinks. Table 4 needs to be cleaned.

5 Structure of the domain

The domain contains all the actions, durative-actions, constants, types, predicates and functions that are required to reach the goal of each problem, respecting the specific requests of the assignment.
In the domain file we have defined 5 types of objects:

- *waiter*: represents a waiter
- *barista*: represents a barista
- *location*: represents a physical location (such as a table or the bar, but also free/occupied spaces for drinks on the tray/waiter's gripper)
- *drinkHot*: represents a hot drink that takes 5-time units to prepare
- *drinkCold*: represents a cold drink that takes 3-time units to prepare

5.1 Predicates

The predicates we used are useful for identifying the location of waiters and drinks, the state of tables (clean or dirty), the waiter's cleaning status, and the state of the drinks (toPrepare, toServe, or consumed). There are also predicates for identifying when a waiter is free and when a table is occupied or assigned to a waiter.

5.2 Functions

In this section we defined the functions for computing the speed of a waiter, the distance between two locations, and the number of drinks, biscuits and consumed drinks at a given location.

5.3 Constants

Constants allow us to not define anytime certain elements that do not change in our problems; we have defined 3 constants for the sake of simplifying the **precondition** of actions and the **init** of goal (of course we could define more constants for improve the simplification).
So we have onTray, which stands for the location whereby a drink is, bar, table1, table2, table3 and table4, which are the places of the coffee shop and finally bartender, that is the only barista which is in charge of preparing drinks.

5.4 Actions

To satisfy all the requirements the following actions have been done:

- *assignTable*: to manage the assignment of the table to one of the two waiters; a waiter can serve/clean only tables at which it is assigned.
- *pickTray*, *dropTry*: actions that allow the waiter to pick and release a tray.
- *loadCold*, *loadHot*: actions to put the hot or cold drinks on the try.
- *grabCold*, *grabHot*: actions to take the hot or cold drinks with the grabber of the waiter.
- *serveColdTry*, *serveHotTry*, *serveCold*, *serveHot*: actions to serve hot and cold drinks from the grabber or from the try.
- *loadBuiscuit*: to put the biscuit on the try.
- *grabBuiscuit*: to take the biscuit with the grabber of the waiter.
- *serveBuiscuitTry*, *serveBiscuit*: actions to serve biscuit from the grabber or from the try.
- *finishConsumationCold*, *finishConsumationHot*: to manage when the hot or cold drinks are consumed
- *customerLeave*: to manage when the customer has finished the consummation and the table needs to be cleaned.

5.5 Durative actions

The following durative-actions are necessary to manage all the requirements:

- *move*: for moving the waiter from one table to another or to the bar
- *prepareHot*, *prepareCold*: allow the barista to prepare the hot or cold drinks, the only difference is the time unit needed.
- *consumeDrinkCold*, *consumeDrinkHot*: for consuming drinks (hot or cold).
- *cleanTable*: action for cleaning table after the customers left.

6 Performance

The performance of our planning is evaluated based on 3 different settings of LPG-TD:

- *n*: specify the total number of solutions that you want to compute. It is also called incremental quality, where the last solution executed is the better one than the previous. Moreover, the number of solution that you want to find is an upper bound, because it can happen that the system never finds the ”n-solution” that you has specified. In this case, the planner stops the execution when it reaches the CPU-time upper bound (that is 1800sec by default).
- *speed*: allows to minimize the research time of the solution. Obviously, this mode is equal to the previous mode when n is set as 1.
- *quality*: is a number that refers to the total number of executed actions. The higher it is, the lower the quality is because we have more actions and the complexity of finding a goal increases.

In addition, you can specify another option, called *-seed*, which sets the seed of the random number generator used by the local search algorithm. Is important to underline that, for instance, one solution that is found with a low search time probably will have a worse quality plan.
One more remarkable thing is that LPG-TD uses a local search algorithm for computing the solution, in particular the **Walkplan** algorithm, which is inspired by the **WalkSat** algorithm. The latter is very important in solving boolean satisfiability problems.
In the following pages, we are making some brief comparisons between the solution to problems that we have found, thanks to some planner options.

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (MOVE W2 TABLE1 TABLE2) [D:0.50; C:1.00]
 0.0000: (ASSIGNTABLE W2 TABLE4) [D:0.00; C:1.00]
 0.0000: (MOVE W1 BAR TABLE3) [D:1.50; C:1.00]
 0.0000: (ASSIGNTABLE W1 TABLE3) [D:0.00; C:1.00]
 0.0000: (PREPARECOLD B D1 TABLE2) [D:3.00; C:1.00]
 0.0000: (ASSIGNTABLE W2 TABLE2) [D:0.00; C:1.00]
 0.5000: (MOVE W2 TABLE2 TABLE4) [D:0.50; C:1.00]
 1.0000: (CLEANTABLE TABLE4 W2) [D:2.00; C:1.00]
 1.5000: (CLEANTABLE TABLE3 W1) [D:4.00; C:1.00]
 3.0000: (PREPARECOLD B D2 TABLE2) [D:3.00; C:1.00]
 5.5000: (MOVE W1 TABLE3 BAR) [D:1.50; C:1.00]
 7.0000: (MOVE W1 BAR TABLE2) [D:1.00; C:1.00]
 8.0000: (MOVE W2 TABLE4 BAR) [D:1.50; C:1.00]
 9.5000: (MOVE W1 TABLE2 TABLE4) [D:0.50; C:1.00]
 9.5000: (GRABCOLD W2 D1 GRAB2) [D:0.00; C:1.00]
 9.5000: (MOVE W2 BAR TABLE3) [D:1.50; C:1.00]
 11.0000: (MOVE W2 TABLE3 TABLE2) [D:0.50; C:1.00]
 11.5000: (SERVECOLD W2 D1 TABLE2 GRAB2) [D:0.00; C:1.00]
 11.5000: (MOVE W2 TABLE2 TABLE3) [D:0.50; C:1.00]
 12.0000: (MOVE W1 TABLE4 TABLE2) [D:0.50; C:1.00]
 12.5000: (MOVE W2 TABLE3 TABLE4) [D:0.50; C:1.00]
 13.0000: (MOVE W2 TABLE4 BAR) [D:1.50; C:1.00]
 14.5000: (PICKTRAY W2 GRAB2) [D:0.00; C:1.00]
 14.5000: (LOADBISCUIT W2 TABLE2 GRAB2) [D:0.00; C:1.00]
 14.5000: (MOVE W2 BAR TABLE1) [D:2.00; C:1.00]
 14.5000: (MOVE W1 TABLE2 TABLE4) [D:0.50; C:1.00]
 15.0000: (MOVE W1 TABLE4 TABLE3) [D:0.50; C:1.00]
 16.5000: (MOVE W2 TABLE1 TABLE2) [D:1.00; C:1.00]
 16.5000: (MOVE W1 TABLE3 BAR) [D:1.50; C:1.00]
 17.5000: (SERVEBISCUITTRAY W2 TABLE2 GRAB2) [D:0.00; C:1.00]
 18.0000: (PICKTRAY W1 GRAB1) [D:0.00; C:1.00]
 18.0000: (LOADCOLD D2 W1 GRAB1) [D:0.00; C:1.00]
 18.0000: (SERVECOLDTRAY W2 D2 TABLE2 GRAB2) [D:0.00; C:1.00]
 18.0000: (LOADBISCUIT W1 TABLE2 GRAB1) [D:0.00; C:1.00]
 18.0000: (SERVEBISCUITTRAY W2 TABLE2 GRAB2) [D:0.00; C:1.00]
 18.0000: (CUSTOMERLEAVE TABLE2) [D:0.00; C:1.00]
 18.0000: (MOVE W1 BAR TABLE3) [D:3.00; C:1.00]
 21.0000: (MOVE W2 TABLE2 BAR) [D:2.00; C:1.00]
 23.0000: (DROPTRAY W2 GRAB2) [D:0.00; C:1.00]
 23.0000: (MOVE W2 BAR TABLE2) [D:1.00; C:1.00]
 24.0000: (CLEANTABLE TABLE2 W2) [D:2.00; C:1.00]


Solution found:
Total time:      0.03
Search time:     0.03
Actions:         41
Duration:        26.000
Plan quality:    41.000
Total Num Flips:  287
```

(a) plan generated in speed modality

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (ASSIGNTABLE W2 TABLE3) [D:0.00; C:1.00]
 0.0000: (MOVE W2 TABLE1 TABLE3) [D:0.50; C:1.00]
 0.0000: (ASSIGNTABLE W2 TABLE4) [D:0.00; C:1.00]
 0.0000: (ASSIGNTABLE W1 TABLE2) [D:0.00; C:1.00]
 0.0000: (PICKTRAY W1 GRAB1) [D:0.00; C:1.00]
 0.0000: (PREPARECOLD B D2 TABLE2) [D:3.00; C:1.00]
 0.5000: (CLEANTABLE TABLE3 W2) [D:4.00; C:1.00]
 3.0000: (LOADCOLD D2 W1 GRAB1) [D:0.00; C:1.00]
 3.0000: (MOVE W1 BAR TABLE2) [D:2.00; C:1.00]
 3.0000: (PREPARECOLD B D1 TABLE2) [D:3.00; C:1.00]
 4.5000: (MOVE W2 TABLE3 TABLE4) [D:0.50; C:1.00]
 5.0000: (CLEANTABLE TABLE4 W2) [D:2.00; C:1.00]
 5.0000: (SERVECOLDTRAY W1 D2 TABLE2 GRAB1) [D:0.00; C:1.00]
 7.0000: (MOVE W2 TABLE4 BAR) [D:1.50; C:1.00]
 8.5000: (PICKTRAY W2 GRAB2) [D:0.00; C:1.00]
 8.5000: (LOADBISCUIT W2 TABLE2 GRAB2) [D:0.00; C:1.00]
 8.5000: (SERVEBISCUITTRAY W1 TABLE2 GRAB1) [D:0.00; C:1.00]
 8.5000: (LOADCOLD D1 W2 GRAB2) [D:0.00; C:1.00]
 8.5000: (SERVECOLDTRAY W1 D1 TABLE2 GRAB1) [D:0.00; C:1.00]
 8.5000: (LOADBISCUIT W2 TABLE2 GRAB2) [D:0.00; C:1.00]
 8.5000: (SERVEBISCUITTRAY W1 TABLE2 GRAB1) [D:0.00; C:1.00]
 8.5000: (CUSTOMERLEAVE TABLE2) [D:0.00; C:1.00]
 8.5000: (MOVE W2 BAR TABLE4) [D:3.00; C:1.00]
 11.5000: (MOVE W1 TABLE2 BAR) [D:2.00; C:1.00]
 13.5000: (DROPTRAY W1 GRAB1) [D:0.00; C:1.00]
 13.5000: (MOVE W1 BAR TABLE2) [D:1.00; C:1.00]
 14.5000: (CLEANTABLE TABLE2 W1) [D:2.00; C:1.00]


Solution number: 6
Total time:      0.16
Search time:     0.16
Actions:         27
Duration:        16.500
Plan quality:    27.000
Total Num Flips: 1688
```

(b) plan generated with incremental quality = 6

Figure 2: Comparison of plans generated in speed and incremental quality mode for problem 1

As shown in the second image of figure 2, this solution has a search-time 5 times higher than the first one, because the planner has performed 5 other searches before the last one. However, in the last case, we have a better quality solution because there are fewer actions executed.

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (MOVE W1 BAR TABLE1) [D:1.00; C:1.00]
 0.0000: (ASSIGNTABLE W1 TABLE1) [D:0.00; C:1.00]
 0.0000: (ASSIGNTABLE W2 TABLE3) [D:0.00; C:1.00]
 0.0000: (PREPARECOLD B D1 TABLE3) [D:3.00; C:1.00]
 1.0000: (CLEANTABLE TABLE1 W1) [D:2.00; C:1.00]
 1.0000: (MOVE W2 TABLE2 BAR) [D:1.00; C:1.00]
 2.0000: (PICKTRAY W2 GRAB2) [D:0.00; C:1.00]
 3.0000: (LOADCOLD D1 W2 GRAB2) [D:0.00; C:1.00]
 3.0000: (MOVE W2 BAR TABLE4) [D:3.00; C:1.00]
 3.0000: (PREPAREHOT B D3 TABLE3) [D:5.00; C:1.00]
 6.0000: (MOVE W2 TABLE4 TABLE3) [D:1.00; C:1.00]
 7.0000: (SERVECOLDTRAY W2 D1 TABLE3 GRAB2) [D:0.00; C:1.00]
 7.0000: (MOVE W2 TABLE3 BAR) [D:3.00; C:1.00]
 8.0000: (PREPARECOLD B D2 TABLE3) [D:3.00; C:1.00]
 10.0000: (LOADBISCUIT W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 10.0000: (MOVE W2 BAR TABLE4) [D:3.00; C:1.00]
 11.0000: (PREPAREHOT B D4 TABLE3) [D:5.00; C:1.00]
 13.0000: (MOVE W2 TABLE4 TABLE2) [D:1.00; C:1.00]
 13.0000: (MOVE W1 TABLE1 BAR) [D:1.00; C:1.00]
 14.0000: (MOVE W2 TABLE2 TABLE3) [D:1.00; C:1.00]
 14.0000: (PICKTRAY W1 GRAB1) [D:0.00; C:1.00]
 14.0000: (LOADHOT D3 W1 GRAB1) [D:0.00; C:1.00]
 15.0000: (SERVEHOTTRAY W2 D3 TABLE3 GRAB2) [D:0.00; C:1.00]
 15.0000: (SERVEBISCUITTRAY W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 15.0000: (LOADBISCUIT W1 TABLE3 GRAB1) [D:0.00; C:1.00]
 15.0000: (SERVEBISCUITTRAY W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 15.0000: (LOADCOLD D2 W1 GRAB1) [D:0.00; C:1.00]
 15.0000: (SERVECOLDTRAY W2 D2 TABLE3 GRAB2) [D:0.00; C:1.00]
 16.0000: (LOADHOT D4 W1 GRAB1) [D:0.00; C:1.00]
 16.0000: (SERVEHOTTRAY W2 D4 TABLE3 GRAB2) [D:0.00; C:1.00]
 16.0000: (LOADBISCUIT W1 TABLE3 GRAB1) [D:0.00; C:1.00]
 16.0000: (SERVEBISCUITTRAY W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 16.0000: (LOADBISCUIT W1 TABLE3 GRAB1) [D:0.00; C:1.00]
 16.0000: (SERVEBISCUITTRAY W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 16.0000: (CUSTOMERLEAVE TABLE3) [D:0.00; C:1.00]
 16.0000: (MOVE W1 BAR TABLE1) [D:2.00; C:1.00]
 18.0000: (MOVE W2 TABLE3 BAR) [D:3.00; C:1.00]
 21.0000: (DROPTRAY W2 GRAB2) [D:0.00; C:1.00]
 21.0000: (MOVE W2 BAR TABLE3) [D:1.50; C:1.00]
 22.5000: (CLEANTABLE TABLE3 W2) [D:4.00; C:1.00]


Solution found:
Total time:      0.03
Search time:     0.02
Actions:         40
Duration:        26.500
Plan quality:    40.000
Total Num Flips:  214
```

(a) plan generated in speed modality

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (ASSIGNTABLE W2 TABLE1) [D:0.00; C:1.00]
 0.0000: (MOVE W2 TABLE2 TABLE1) [D:0.50; C:1.00]
 0.0000: (PICKTRAY W1 GRAB1) [D:0.00; C:1.00]
 0.0000: (PREPARECOLD B D2 TABLE3) [D:3.00; C:1.00]
 0.0000: (ASSIGNTABLE W1 TABLE3) [D:0.00; C:1.00]
 0.5000: (CLEANTABLE TABLE1 W2) [D:2.00; C:1.00]
 3.0000: (LOADCOLD D2 W1 GRAB1) [D:0.00; C:1.00]
 3.0000: (MOVE W1 BAR TABLE3) [D:3.00; C:1.00]
 3.0000: (PREPAREHOT B D4 TABLE3) [D:5.00; C:1.00]
 6.0000: (SERVECOLDTRAY W1 D2 TABLE3 GRAB1) [D:0.00; C:1.00]
 6.0000: (MOVE W2 TABLE1 BAR) [D:1.00; C:1.00]
 7.0000: (PICKTRAY W2 GRAB2) [D:0.00; C:1.00]
 7.0000: (LOADBISCUIT W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 7.0000: (SERVEBISCUITTRAY W1 TABLE3 GRAB1) [D:0.00; C:1.00]
 8.0000: (LOADHOT D4 W2 GRAB2) [D:0.00; C:1.00]
 8.0000: (SERVEHOTTRAY W1 D4 TABLE3 GRAB1) [D:0.00; C:1.00]
 8.0000: (LOADBISCUIT W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 8.0000: (SERVEBISCUITTRAY W1 TABLE3 GRAB1) [D:0.00; C:1.00]
 8.0000: (PREPAREHOT B D3 TABLE3) [D:5.00; C:1.00]
 13.0000: (LOADHOT D3 W2 GRAB2) [D:0.00; C:1.00]
 13.0000: (SERVEHOTTRAY W1 D3 TABLE3 GRAB1) [D:0.00; C:1.00]
 13.0000: (LOADBISCUIT W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 13.0000: (SERVEBISCUITTRAY W1 TABLE3 GRAB1) [D:0.00; C:1.00]
 13.0000: (PREPARECOLD B D1 TABLE3) [D:3.00; C:1.00]
 16.0000: (LOADCOLD D1 W2 GRAB2) [D:0.00; C:1.00]
 16.0000: (SERVECOLDTRAY W1 D1 TABLE3 GRAB1) [D:0.00; C:1.00]
 16.0000: (LOADBISCUIT W2 TABLE3 GRAB2) [D:0.00; C:1.00]
 16.0000: (SERVEBISCUITTRAY W1 TABLE3 GRAB1) [D:0.00; C:1.00]
 16.0000: (CUSTOMERLEAVE TABLE3) [D:0.00; C:1.00]
 16.0000: (MOVE W2 BAR TABLE2) [D:2.00; C:1.00]
 18.0000: (MOVE W1 TABLE3 BAR) [D:3.00; C:1.00]
 21.0000: (DROPTRAY W1 GRAB1) [D:0.00; C:1.00]
 21.0000: (MOVE W1 BAR TABLE3) [D:1.50; C:1.00]
 22.5000: (CLEANTABLE TABLE3 W1) [D:4.00; C:1.00]


Solution number: 5
Total time:      0.38
Search time:     0.38
Actions:         34
Duration:        26.500
Plan quality:    34.000
Total Num Flips: 2869
```

(b) plan generated with incremental quality = 5

Figure 3: Comparison of plans generated in speed and incremental quality mode for problem 2

Since the target has more predicates to execute obviously the complexity of the plan increases, and therefore it will take more time to find the final solution. It is important to note that in the solution the number of incremental quality maximum executable is 5, after which it can not find the solution. This means that it is not possible to find a 6 solution that is better than the previous ones within 1800 seconds.

```
 10.0000: (MOVE W1 BAR TABLE2) [D:1.00; C:1.00]
 10.0000: (PREPAREHOT B D2 TABLE1) [D:5.00; C:1.00]
 11.0000: (MOVE W1 TABLE2 TABLE4) [D:0.50; C:1.00]
 11.0000: (MOVE W2 TABLE3 BAR) [D:1.50; C:1.00]
 11.5000: (MOVE W1 TABLE4 TABLE1) [D:0.50; C:1.00]
 12.5000: (MOVE W2 BAR TABLE4) [D:1.50; C:1.00]
 14.0000: (MOVE W1 TABLE1 BAR) [D:1.00; C:1.00]
 15.0000: (MOVE W2 TABLE4 TABLE1) [D:0.50; C:1.00]
 15.0000: (PICKTRAY W1 GRAB1) [D:0.00; C:1.00]
 15.0000: (LOADHOT D4 W1 GRAB1) [D:0.00; C:1.00]
 15.0000: (PREPAREHOT B D3 TABLE4) [D:5.00; C:1.00]
 15.5000: (MOVE W1 BAR TABLE4) [D:3.00; C:1.00]
 18.5000: (MOVE W2 TABLE1 BAR) [D:1.00; C:1.00]
 19.5000: (PICKTRAY W2 GRAB2) [D:0.00; C:1.00]
 19.5000: (LOADHOT D1 W2 GRAB2) [D:0.00; C:1.00]
 19.5000: (MOVE W2 BAR TABLE2) [D:2.00; C:1.00]
 21.5000: (MOVE W2 TABLE2 TABLE1) [D:1.00; C:1.00]
 22.5000: (SERVEHOTTRAY W2 D1 TABLE1 GRAB2) [D:0.00; C:1.00]
 22.5000: (MOVE W2 TABLE1 BAR) [D:2.00; C:1.00]
 24.5000: (LOADBISCUIT W2 TABLE1 GRAB2) [D:0.00; C:1.00]
 24.5000: (LOADHOT D2 W2 GRAB2) [D:0.00; C:1.00]
 24.5000: (MOVE W2 BAR TABLE1) [D:2.00; C:1.00]
 26.5000: (SERVEBISCUITTRAY W2 TABLE1 GRAB2) [D:0.00; C:1.00]
 26.5000: (SERVEHOTTRAY W2 D2 TABLE1 GRAB2) [D:0.00; C:1.00]
 26.5000: (MOVE W2 TABLE1 BAR) [D:2.00; C:1.00]
 28.5000: (LOADBISCUIT W2 TABLE1 GRAB2) [D:0.00; C:1.00]
 28.5000: (MOVE W2 BAR TABLE1) [D:2.00; C:1.00]
 30.5000: (SERVEBISCUITTRAY W2 TABLE1 GRAB2) [D:0.00; C:1.00]
 30.5000: (CUSTOMERLEAVE TABLE1) [D:0.00; C:1.00]
 30.5000: (MOVE W2 TABLE1 BAR) [D:2.00; C:1.00]
 32.5000: (MOVE W2 BAR TABLE2) [D:2.00; C:1.00]
 32.5000: (MOVE W1 TABLE4 TABLE1) [D:1.00; C:1.00]
 34.5000: (MOVE W2 TABLE2 TABLE4) [D:1.00; C:1.00]
 35.5000: (SERVEHOTTRAY W2 D4 TABLE4 GRAB2) [D:0.00; C:1.00]
 35.5000: (MOVE W2 TABLE4 BAR) [D:3.00; C:1.00]
 38.5000: (LOADBISCUIT W2 TABLE4 GRAB2) [D:0.00; C:1.00]
 38.5000: (DROPTRAY W2 GRAB2) [D:0.00; C:1.00]
 38.5000: (MOVE W1 TABLE1 TABLE4) [D:1.00; C:1.00]
 39.5000: (MOVE W2 BAR TABLE1) [D:1.00; C:1.00]
 40.5000: (CLEANTABLE TABLE1 W2) [D:2.00; C:1.00]
 40.5000: (MOVE W1 TABLE4 BAR) [D:3.00; C:1.00]
 43.5000: (MOVE W1 BAR TABLE4) [D:3.00; C:1.00]
 46.5000: (MOVE W1 TABLE4 BAR) [D:3.00; C:1.00]
 49.5000: (MOVE W1 BAR TABLE3) [D:3.00; C:1.00]
 52.5000: (MOVE W2 TABLE1 BAR) [D:1.00; C:1.00]
 53.5000: (PICKTRAY W2 GRAB2) [D:0.00; C:1.00]
 53.5000: (MOVE W2 BAR TABLE4) [D:3.00; C:1.00]
 56.5000: (SERVEBISCUITTRAY W2 TABLE4 GRAB2) [D:0.00; C:1.00]
 56.5000: (MOVE W1 TABLE3 BAR) [D:3.00; C:1.00]
 59.5000: (LOADHOT D3 W1 GRAB1) [D:0.00; C:1.00]
 59.5000: (SERVEHOTTRAY W2 D3 TABLE4 GRAB2) [D:0.00; C:1.00]
 59.5000: (LOADBISCUIT W1 TABLE4 GRAB1) [D:0.00; C:1.00]
 59.5000: (SERVEBISCUITTRAY W2 TABLE4 GRAB2) [D:0.00; C:1.00]
 59.5000: (MOVE W1 BAR TABLE1) [D:2.00; C:1.00]
 59.5000: (CUSTOMERLEAVE TABLE4) [D:0.00; C:1.00]
 61.5000: (MOVE W2 TABLE4 BAR) [D:3.00; C:1.00]
 64.5000: (DROPTRAY W2 GRAB2) [D:0.00; C:1.00]
 64.5000: (MOVE W2 BAR TABLE4) [D:1.50; C:1.00]
 66.0000: (CLEANTABLE TABLE4 W2) [D:2.00; C:1.00]


Solution found:
Total time:      0.04
Search time:     0.04
Actions:         73
Duration:        68.000
Plan quality:    73.000
Total Num Flips:  537
```

(a) plan generated in speed modality

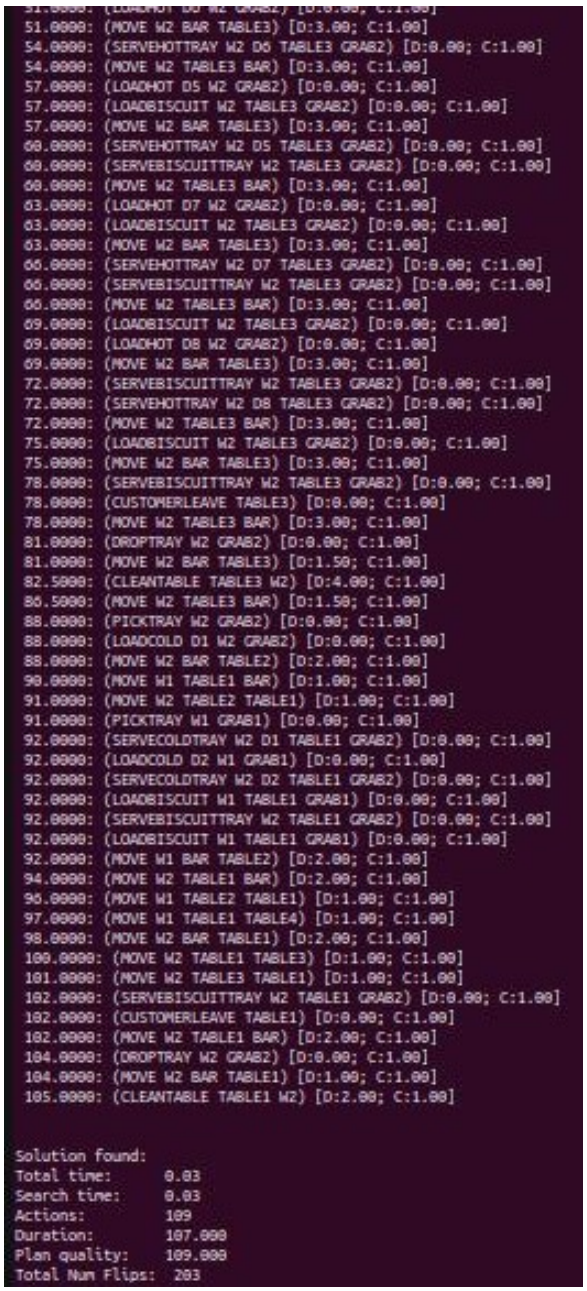
```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (ASSIGNTABLE W1 TABLE1) [D:0.00; C:1.00]
 0.0000: (PICKTRAY W1 GRAB1) [D:0.00; C:1.00]
 0.0000: (MOVE W1 BAR TABLE3) [D:3.00; C:1.00]
 0.0000: (PREPAREHOT B D2 TABLE1) [D:5.00; C:1.00]
 0.0000: (ASSIGNTABLE W2 TABLE3) [D:0.00; C:1.00]
 0.0000: (ASSIGNTABLE W2 TABLE4) [D:0.00; C:1.00]
 3.0000: (MOVE W1 TABLE3 BAR) [D:3.00; C:1.00]
 5.0000: (PREPAREHOT B D1 TABLE1) [D:5.00; C:1.00]
 6.0000: (LOADHOT D2 W1 GRAB1) [D:0.00; C:1.00]
 6.0000: (MOVE W1 BAR TABLE1) [D:2.00; C:1.00]
 6.0000: (MOVE W2 TABLE2 TABLE3) [D:0.50; C:1.00]
 6.5000: (CLEANTABLE TABLE3 W2) [D:4.00; C:1.00]
 8.0000: (SERVEHOTTRAY W1 D2 TABLE1 GRAB1) [D:0.00; C:1.00]
 8.0000: (MOVE W1 TABLE1 BAR) [D:2.00; C:1.00]
 10.0000: (LOADHOT D1 W1 GRAB1) [D:0.00; C:1.00]
 10.0000: (LOADBISCUIT W1 TABLE1 GRAB1) [D:0.00; C:1.00]
 10.0000: (MOVE W1 BAR TABLE1) [D:2.00; C:1.00]
 10.0000: (PREPAREHOT B D4 TABLE4) [D:5.00; C:1.00]
 12.0000: (SERVEHOTTRAY W1 D1 TABLE1 GRAB1) [D:0.00; C:1.00]
 12.0000: (SERVEBISCUITTRAY W1 TABLE1 GRAB1) [D:0.00; C:1.00]
 12.0000: (MOVE W1 TABLE1 BAR) [D:2.00; C:1.00]
 14.0000: (LOADBISCUIT W1 TABLE1 GRAB1) [D:0.00; C:1.00]
 14.0000: (MOVE W1 BAR TABLE1) [D:2.00; C:1.00]
 15.0000: (PREPAREHOT B D3 TABLE4) [D:5.00; C:1.00]
 16.0000: (SERVEBISCUITTRAY W1 TABLE1 GRAB1) [D:0.00; C:1.00]
 16.0000: (MOVE W1 TABLE1 BAR) [D:2.00; C:1.00]
 16.0000: (CUSTOMERLEAVE TABLE1) [D:0.00; C:1.00]
 18.0000: (DROPTRAY W1 GRAB1) [D:0.00; C:1.00]
 18.0000: (MOVE W1 BAR TABLE1) [D:1.00; C:1.00]
 19.0000: (CLEANTABLE TABLE1 W1) [D:2.00; C:1.00]
 19.0000: (MOVE W2 TABLE3 BAR) [D:1.50; C:1.00]
 20.5000: (PICKTRAY W2 GRAB2) [D:0.00; C:1.00]
 20.5000: (MOVE W2 BAR TABLE4) [D:3.00; C:1.00]
 23.5000: (MOVE W1 TABLE1 BAR) [D:1.00; C:1.00]
 24.5000: (PICKTRAY W1 GRAB1) [D:0.00; C:1.00]
 24.5000: (LOADHOT D4 W1 GRAB1) [D:0.00; C:1.00]
 24.5000: (SERVEHOTTRAY W2 D4 TABLE4 GRAB2) [D:0.00; C:1.00]
 24.5000: (LOADBISCUIT W1 TABLE4 GRAB1) [D:0.00; C:1.00]
 24.5000: (SERVEBISCUITTRAY W2 TABLE4 GRAB2) [D:0.00; C:1.00]
 24.5000: (LOADHOT D3 W1 GRAB1) [D:0.00; C:1.00]
 24.5000: (SERVEHOTTRAY W2 D3 TABLE4 GRAB2) [D:0.00; C:1.00]
 24.5000: (LOADBISCUIT W1 TABLE4 GRAB1) [D:0.00; C:1.00]
 24.5000: (SERVEBISCUITTRAY W2 TABLE4 GRAB2) [D:0.00; C:1.00]
 24.5000: (CUSTOMERLEAVE TABLE4) [D:0.00; C:1.00]
 24.5000: (MOVE W1 BAR TABLE3) [D:3.00; C:1.00]
 27.5000: (MOVE W2 TABLE4 BAR) [D:3.00; C:1.00]
 30.5000: (DROPTRAY W2 GRAB2) [D:0.00; C:1.00]
 30.5000: (MOVE W2 BAR TABLE4) [D:1.50; C:1.00]
 32.0000: (CLEANTABLE TABLE4 W2) [D:2.00; C:1.00]


Solution number: 6
Total time:      2.19
Search time:     2.18
Actions:         49
Duration:        34.000
Plan quality:    49.000
Total Num Flips: 10525
```

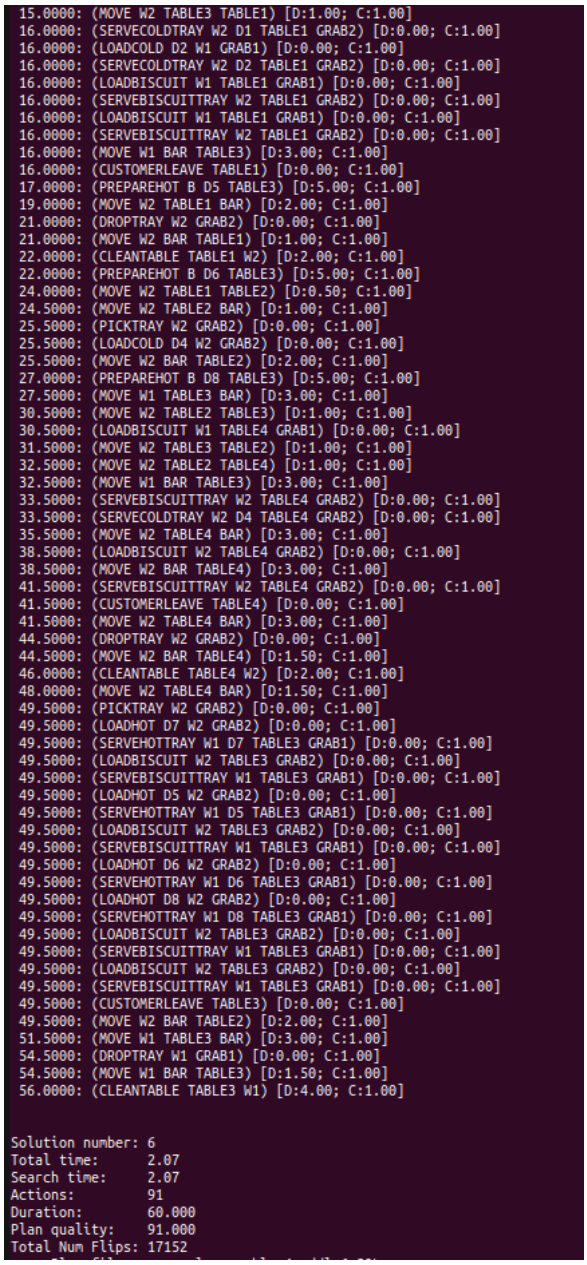
(b) plan generated with incremental quality = 6

Figure 4: Comparison of plans generated in speed and incremental quality mode for problem 3

In the problem, we have more objects to take into account. This means that the number of actions is higher and also the number referred to the quality of the plan; as a consequence, the quality of the first image in figure 4 is worse than the previous two shown in figure 3.



(a) plan generated in speed modality



(b) plan generated with incremental quality = 6

Figure 5: Comparison of plans generated in speed and incremental quality mode for problem 4

Although we expected a solution with a higher search time than the previous problem (because in the goal we have more predicates and objects), we still obtained a lower search time because the seed option executes the plan with the specified random number generator; without that option, the situation reverses.

notes:

- When running the planner in incremental quality mode with high numbers (such as 5 or 6), you will get repeatedly a warning, saying that the problem should be run with the '-temporal 1' option'; you can ignore the warning, since the solution to the problem will be found anyway, after some time.

7 Improvements

As it was said previously, using the LPG++/LPG-TD planners, it was not possible to model the fact that hot drinks cool down in four-time units.
To better explain the problem we have encountered, we will first describe the implementation we had thought for this situation:

- After the *prepareHot* durative-action, the drink would normally be put on the bar and ready to be loaded by one of the two waiters; to make that the cooling down process would start, we would have inserted a new precondition in the *serveHot* and *serveHotTray* actions, called **cooling-down**;

- This new precondition, would be set true **at start** of the *coolingDown* durative-action; in this way, a hot drink must be cooling down while it is picked and served by a waiter.
  This idea was adopted to force the planner to do this durative-action: in fact, differently from a process, a durative-action does not start automatically when its preconditions are satisfied, so we had to find a way to force the planner to execute it as if it was an exogenous event;

- To conclude, another predicate would be inserted as affect **at end** of the *coolingDown* durative-action, to alert the planner that the hot drink has cooled down and the customer will not accept it, so the drink needs to be tossed and a new one has to be made.

The problem we have encountered is the following: when a durative-action starts, it does not activate its **at start** effects. Instead, those effects are activated **at end**. This makes it impossible to be in a state where the predicate **cooling-down** is true and the predicate **cooled** is not, because the durative-action *coolingDown* will make them both true at the same time. This problem appears only here, because this is the only situation in this problem where we have to execute an action (*serveHot* or *serveHotTray*) while a durative-action is executing.
This problem could have been solved by modelling the **coolingDown** situation as a process, but, as it was already said, another problem emerged: the LPG planners do not support concurrency between processes and durative-actions.

List of Figures