ARTIFICIAL INTELLIGENCE FOR ROBOTICS II

Assignment 1

AI Planning

Group K

Hafiz Muizz Ahmed Sethi (5428151),
Naveed Manzoor Afridi (5149575),
Raja Farooq (5245187),
Furqan Ansari (5197073)

Submitted to
Prof. Mauro Vallati

Preface

The work is based on AI Planning which involves the demonstration of an automated warehouse for a company. An automated warehouse helps in order management in a precise way which automatically boost the storage capacity.

Acknowledgements

We take this opportunity to express our sincere thanks to Prof. Mauro Vallati for their guidance and support. We would also like to express our gratitude towards him for showing confidence in us. It was a privilege to have a great experience working under him in a cordial environment. We are very much thankful to the University of Genoa, for providing us the opportunity of pursue M.Sc in Robotics Engineering in a peaceful environment with ample resources. In the end, we would like to acknowledge our parents, friends. Without their support, this work would not have been possible.

Abstract

This assignment is about AI Planning in which the task is to model a robotic coffee shop scenario in an efficient manner. We presented with some techniques for handling planning problem throughout the course such as PDDL+/ Numeric planning, Numeric and Temporal planning, and so on. For this assignment, we used Numeric and Temporal planning using LPG 1.2 Planner. A planner is used to design an application domain and problem (defines the initial state and goal state), and also to generate a plan.

Contents

1 Introduction

The task is to automate a coffee shop, by AI Planning. At coffee shop there are two robots which perform action to prepare and serve the drinks to customer. The barista robot needs to prepare the drinks ordered by the customers. Cold drinks are faster to prepare than warm drinks. The waiter robot is then in charge of serving customers, and cleaning tables where customers have already left the shop. The waiter robot can decide to grasp a drink with its gripper, or to use a tray to carry more drinks at once. It takes the barista robot 3-time units to prepare a cold drink, and 5-time units to prepare a warm drink. Once ready, the drinks are put in the bar, where the waiter can pick them up. The waiter robot can grasp a single drink using one of its grippers and bring it to the table where the corresponding customer is seated. The waiter is not able to grasp one drink per each gripper but can only bring a drink at a time if it is not using a tray. The tray can be taken from the bar and must be returned there after use. The waiter is not allowed to leave the tray on a table. The waiter moves at 2 meters per time unit; 1 meter per time unit if it is using the tray. Finally, the robot has to clean tables: it takes 2-time units per square meter to clean a table. The robot cannot clean a table while carrying the tray.

2 Related Work

Fox M. at al discussed the syntax and modelling style of pddl+, showing that the languagemakes the modelling of complex time-dependent effects an effective one [1].

Cushing, W. demonstrated by way of compilation methods and analysis of algorithmic properties such as completeness that the more immediately pressing computational obstacles can be dealt with in a more efficient manner [2].

Li H. at al discussed that the Autonomous underwater vehicles (AUV) paradigm, however, requires robust operation that is cost effective and responsive to the environment and its major challenge is to sense remotely [3].

Penberthy S. at al present ZENO, a least commitment planner that handles actions occurring over extended in- tervals of time. Deadline goals, metric preconditions, metric effects, and continuous change are supported [4].

Vallati M. at al discussed about the ParLPG planning system is based on the idea of using a generic algorithm configuration procedure – here, the well-known ParamILS framework – to opti- mise the performance of a highly parametric planner on a set of problem instances representative of a specific planning domain [5].

Long D. at al analyzed the resultsfrom several perspectives, in order to address the questions of comparative performance between planners, comparative difficulty of domains, the degree of agreement between planners about the relative difficulty of individual problem instances and the question of how well planners scale relative to one another over increasingly difficult problems[6].

McDermott at al developed a first-order temporal logic in which it is possible to prove and name things about facts and events.To be specific, the logic provides analyses of casualty [7].

Smith D. at al discusse on the paper about TGP, a new algorithm for temporal planning. TGP operates by incrementally expanding a compact planning graph representation that handles actions of differing duration [8].

Vila L. discussed the Temporal Reasoning. Then, the most important representational issues which determine a Temporal Reasoning approach are introduced: the logical form on which the approach is based, the ontology and the concepts related with reasoning about action [9].

Bacchus F. at al discussed how domain-dependent search control knowledge can be represented in a temporal logic, and then utilized to effectively control a forward-chaining planner[10].

3 Methodology

This section will describes the methodology and the approach used for this assignment to accomplish a goal. We produces a single PDDL domain file and the corresponding planning problem files. We used Numeric and temporal Planning technique to achieve the goal. In each case we are given with the order details that customer will order i.e. cold or warm drink, which table have to be cleaned etc. Which robot will have to prepare drink and which one will serve.

To design a plan we used a LPG Planner. LPG (Local search for Planning Graphs) is a planner which is based on local search and planning graphs. These will grasps PDDL 2.1 domains. This PDDL 2.1 domains associates with numerical quantities and durations. This planner able to solve not only plan generation but also plan adaptation problems. The evaluation function uses some heuristics to estimate the "search cost" and the "execution cost" of achieving a (possibly numeric) precondition. Action durations and numerical quantities (e.g., fuel consumption) are represented in the actions graphs, and are modeled in the evaluation function. In temporal domains, actions are ordered using a "precedence graph" that is maintained during search, and that takes into account the mutex relations of the planning graph. The system can produce good quality plans in terms of one or more criteria. This is achieved by an anytime process producing a sequence of plans, each of which is an improvement of the previous ones in terms of its quality. LPG is integrated with a best-first algorithm similar to the one used by FF. The system can automatically switch to best-first search after a certain number of search steps and "restarts" have been performed. Finally, LPG can be used as a pre-processor to produce a quasi-solution that is then repaired by ADJ, a plan-analysis technique for fast plan-adaptation (*source from: https://lpg.unibs.it/lpg/*).

Next, we describes the syntactic foundations of our code. For this, we created single domain file and comparable four problem files. A "domain file" in PDDL depicts the "universal" features of a problem. These universal features will not interchange anyway of what the case we are trying to solve. In domain file, there are mainly requirements, object types, predicates, functions, and actions which can exist within the model design. General structure of a domain is as follows:

```
(define (domain domainname)
(:requirements [:strips] [:typing] [:adl])
(:predicates (predicatename11 [?X1 ?X2 ... ?Xn])(predicatename2
[?Y1 ?Y2 ... ?Yn])
...)
(:action actionname

[:parameters (?L1 ?L2 ... ?Ln)]
[:precondition preconditionformula]
[:effect effectformula]
)
...)
```

The "problem file" in PDDL describes the other part of the planning. The problem consists ofobjects, init (initial stage), and goal. General structure of problem file is as follows:

```
(define (problem porblemname)(:domain
domainname)
(:objects O1 O2......... On)
(:init I1 I2 .........In)
(:goal conditionformula)
)
```

In our assignment, we worked on LPG Planner which is based on PDDL 2.1 and PDDL 2.1 was language of the 2002 AIPS Competition and construct on the syntax defined for PDDL 1.2 (*source from: https://planning.wiki/ref/pddl21*).

Domain file in PDDL 2.1, is the extended version of PDDL 1.2 which contains two new attributes such as durative-actions and functions. These two new features are raise to numeric fluent. General structure of domain in PDDL 2.1 is as follows:

```
(define (domain domain-name)
(:requirements :durative-actions :fluents :duration-inequalities)
(:types t1 t2 .......tn)
(:predicates
...
) v (:functions
(functions - f1)
(functions- f2)
.....
(functions -fn) v )
(:durative-action actionname
:parameters (argument)
:duration (= ?duration durationnumber)
:condition (expression of condition)
:effect (expression of effect)
)
)
```

Problem file in PDDL 2.1, is the expanded in conductive to complement the syntax of the domain file. In PDDL 2.1, metric is the additional features is added to the problem file. Also, metric feature acts like an optimal function role which depicts the cost value for a designed plan.

```
(define
(problem problem-name)
(:domain domain-name)
(:objects
O1 O2 .......On - objects
)
(:init
```

(initial state)
...
) v (:goal (goal state
...
)
)
(:metric (numericexpreation)
)
)
)

In this part we will describe about a description of the meaning of each component of the modelused in our code.

Domain File: It comprises of

(i) *Requirements:* Added the requirements for numeric and temporal planning.

(:requirements :strips :typing :durative-actions :numeric-fluents)

(ii) *Types:* Defined the types of objects to differentiate them such as robot, drink, customer, table. Also, distinguish among the drink like warm-drink and cold-drink

(iii) *Predicates:* This will highlights predicates based on our task such as:

- Order-placed: tell us which drink is order by customers.
- Drink-ready : tell us whether the drink is prepared by robot or not.
- Drink-served: carry the drink to customers sits on the table.
- Tray-in-use and tray-with-drinks; describe about the drink is on tray or not.
- Cleaned: describe whether the desire table is cleaned by robot.

(iv) *Durative-Actions:* This will defines the actions taken by the robots throughout the plan

- *Prepare cold-drink/warm-drink: this action define that the robot prepare the warm/cold drink.*
- *Pickup drink: it define that the drink is ready to served to customers using a gripper or tray.*
- Pickup-serve-return: define that tray should be picked from the bar counter and drink will pe placed on it, served to customers and then return back to the bar counter.
- Clean-table : this action define the robot should have to clean the table.

Problem File: This part consists of

(i) Objects: The objects which are defined in the problem file represents that what objects to be encoded in the problem file. There are different variables declared to distinguish among the objects such as: warm-drink and cold-drink, table-1, table-2…table-4 and also customer1 etc.

(ii) Init: This section is the initial state of the plan. This will highlights the initial position of the objects such as

Problem 1: In this problem, there are 2 customers, at table-2 and they order 2 cold-drink . Also, table-3 and table-4 needed to be cleaned.

Problem2 : There are 4 customers at table 3: they ordered 2 cold drinks and 2 warm drinks. Table 1 needs to be cleaned.

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.

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.

(iii)Goal state: this state will define the goal state of the planning model. In this the Drinks are served to customers on the desired table also tray return to the bar by the robot and then the table is cleaned by the r

4 Results

This section will describes the experimental results which shows the LPG performance using Numeric and temporal Planning problems. The production of LPG was carryout in terms not only CPU-time required to find a results but also the quality of plan computed. This will also defines that how the robots actions to prepare and carry the drinks according to accomplishing the task.

Here are the some results parameters obtain from the planners such as:

Modality: Incremental Planner

	Number of Action	Number of Conditional Actions	Number of Facts
Problem 1	**48**	**0**	**83**
Problem 2	**116**	**0**	**215**
Problem 3	**116**	**0**	**215**
Problem 4	**188**	**0**	**351**

Figure 4.1: Actions to solve a problems

The above figure highlights the table that describes the actions taken by the planer to solvethe problems.

Analyzing Planning Problem:

(v) Temporal Planning Problem: YES

(vi) Numeric Planning Problem: YES

(vii) Problem with Timed Initial Literals: NO

(viii)Problem with Derived Predicates: NO

Evaluation function weights:

(i) Action duration 1.00

(ii) Action cost 0.00

Below are the figures which shows the results after the plan computed.

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (CLEAN-TABLE WAITER TABLE-3) [D:1.00; C:1.00]
 0.0000: (CLEAN-TABLE WAITER TABLE-4) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-1) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-2) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK) [D:1.00; C:1.00]
 1.0000: (PICKUP-DRINK WAITER COLD-DRINK-2) [D:1.00; C:1.00]
 2.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-2) [D:1.00; C:1.00]
 3.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-2 COLD-DRINK-2 TABLE-2 MAIN) [D:1.00; C:3.00]
 4.0000: (PICKUP-DRINK WAITER COLD-DRINK) [D:1.00; C:1.00]
 5.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-1 COLD-DRINK-1 TABLE-2 MAIN) [D:1.00; C:3.00]
```

Figure 4.2: Problem 1 plan computed result

Figure 4.2 shows the results after the plan computation of Problem 1. It highlights the action duration i.e. action cost of the plan. Also, describes the plan execution such as robot performance to accomplish the goal.

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (CLEAN-TABLE WAITER TABLE-1) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK1) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK2) [D:1.00; C:1.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK2) [D:1.00; C:2.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK1) [D:1.00; C:2.00]
 1.0000: (PICKUP-DRINK WAITER COLD-DRINK2) [D:1.00; C:1.00]
 2.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK2) [D:1.00; C:1.00]
 3.0000: (PICK-SERVE-RETURN BARISTA WAITER C2 COLD-DRINK2 TABLE-3 MAIN) [D:1.00; C:3.00]
 4.0000: (PICKUP-DRINK WAITER WARM-DRINK2) [D:1.00; C:1.00]
 5.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK2) [D:1.00; C:2.00]
 6.0000: (PICK-SERVE-RETURN BARISTA WAITER C4 WARM-DRINK2 TABLE-3 MAIN) [D:1.00; C:3.00]
 7.0000: (PICKUP-DRINK WAITER WARM-DRINK1) [D:1.00; C:1.00]
 8.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK1) [D:1.00; C:2.00]
 9.0000: (PICK-SERVE-RETURN BARISTA WAITER C3 WARM-DRINK1 TABLE-3 MAIN) [D:1.00; C:3.00]
 10.0000: (PICKUP-DRINK WAITER COLD-DRINK) [D:1.00; C:1.00]
 11.0000: (PICK-SERVE-RETURN BARISTA WAITER C1 COLD-DRINK1 TABLE-3 MAIN) [D:1.00; C:3.00]
```

Figure 4.3: Problem 2 plan computed result

Figure 4.3 shows the results after the plan computation of Problem 2. It highlights the action duration i.e. action cost of the plan. Also, describes the plan execution such as robots performance to accomplish the goal.

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (CLEAN-TABLE WAITER TABLE-3) [D:1.00; C:1.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-4) [D:1.00; C:2.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK) [D:1.00; C:1.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-3) [D:1.00; C:2.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-2) [D:1.00; C:2.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-1) [D:1.00; C:2.00]
 1.0000: (PICKUP-DRINK WAITER WARM-DRINK-2) [D:1.00; C:1.00]
 2.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-3 WARM-DRINK-3 TABLE-1 MAIN) [D:1.00; C:3.00]
 2.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-2) [D:1.00; C:2.00]
 3.0000: (PICKUP-DRINK WAITER WARM-DRINK-2) [D:1.00; C:1.00]
 4.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-2) [D:1.00; C:2.00]
 5.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-2 WARM-DRINK-2 TABLE-4 MAIN) [D:1.00; C:3.00]
 6.0000: (PICKUP-DRINK WAITER WARM-DRINK-1) [D:1.00; C:1.00]
 7.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-1) [D:1.00; C:2.00]
 8.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-1 WARM-DRINK-1 TABLE-4 MAIN) [D:1.00; C:3.00]
 9.0000: (PICKUP-DRINK WAITER COLD-DRINK) [D:1.00; C:1.00]
 10.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-4 WARM-DRINK-4 TABLE-1 MAIN) [D:1.00; C:3.00]
```

Figure 4.4: Problem 3 plan computed result

Figure 4.4 shows the results after the plan computation of Problem 3. It highlights the action duration i.e. action cost of the plan. Also, describes the plan execution such as robots performance to accomplish the goal.

```
Plan computed:
   Time: (ACTION) [action Duration; action Cost]
 0.0000: (CLEAN-TABLE WAITER TABLE-4) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-1) [D:1.00; C:1.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-4) [D:1.00; C:2.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-4) [D:1.00; C:1.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-2) [D:1.00; C:2.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-2) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-3) [D:1.00; C:1.00]
 0.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK) [D:1.00; C:1.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-1) [D:1.00; C:2.00]
 0.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-3) [D:1.00; C:2.00]
 1.0000: (PICKUP-DRINK WAITER COLD-DRINK-4) [D:1.00; C:1.00]
 2.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-4) [D:1.00; C:1.00]
 2.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-4 WARM-DRINK-4 TABLE-3 MAIN) [D:1.00; C:3.00]
 3.0000: (PICKUP-DRINK WAITER WARM-DRINK-2) [D:1.00; C:1.00]
 3.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-4) [D:1.00; C:2.00]
 4.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-4 COLD-DRINK-4 TABLE-1 MAIN) [D:1.00; C:3.00]
 4.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-2) [D:1.00; C:2.00]
 5.0000: (PICKUP-DRINK WAITER WARM-DRINK-2) [D:1.00; C:1.00]
 6.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-2) [D:1.00; C:2.00]
 7.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-2 WARM-DRINK-2 TABLE-3 MAIN) [D:1.00; C:3.00]
 8.0000: (PICKUP-DRINK WAITER COLD-DRINK-2) [D:1.00; C:1.00]
 9.0000: (PREPARE-COLD-DRINK BARISTA COLD-DRINK-2) [D:1.00; C:1.00]
 10.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-2 COLD-DRINK-2 TABLE-4 MAIN) [D:1.00; C:3.00]
 11.0000: (PICKUP-DRINK WAITER COLD-DRINK) [D:1.00; C:1.00]
 12.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-3 COLD-DRINK-3 TABLE-1 MAIN) [D:1.00; C:3.00]
 13.0000: (PICKUP-DRINK WAITER WARM-DRINK-1) [D:1.00; C:1.00]
 14.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-1) [D:1.00; C:2.00]
```

```
 15.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-1 WARM-DRINK-1 TABLE-3 MAIN) [D:1.00; C:3.00]
 16.0000: (PICKUP-DRINK WAITER WARM-DRINK-3) [D:1.00; C:1.00]
 17.0000: (PREPARE-WARM-DRINK BARISTA WARM-DRINK-3) [D:1.00; C:2.00]
 18.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-3 WARM-DRINK-3 TABLE-3 MAIN) [D:1.00; C:3.00]
 19.0000: (PICKUP-DRINK WAITER WARM-DRINK-4) [D:1.00; C:1.00]
 20.0000: (PICK-SERVE-RETURN BARISTA WAITER CUSTOMER-1 COLD-DRINK-1 TABLE-4 MAIN) [D:1.00; C:3.00]
```

Figure 4.5: Problem 4 plan computed result

Figure 4.5 shows the results after the plan computation of Problem 4. It highlights the action duration i.e. action cost of the plan. Also, describes the plan execution such a robots performance to accomplish the goal.

5 Conclusion

In conclusion, the scenario of using robots to run a coffee shop highlights the increasing trend of automation and robotics in various industries. The use of robots in such scenarios can improve efficiency, reduce errors, and enhance the overall customer experience. However, as with any new technology, careful planning and design are essential to ensure optimal performance, reliability, and safety. This scenario also underscores the importance of interdisciplinary collaboration, involving experts from different fields such as robotics, artificial intelligence, human factors, and design. As we continue to move towards a more automated future, such examples of robotic coffee shops provide valuable insights into the potential benefits and challenges of using robots in various industries

6 References

[1] Fox M.; Long D.(2006). Modelling Mixed Discrete-Continuous Domains for Planning. Journal of Artificial Intelligence Research 27 235–297.

[2] Cushing, W. (2012). When is temporal planning really temporal? PhD Thesis, Arizona State University.

[3] Li, H., and Williams, B. (2011). Hybrid planning with temporally extended goals for sustainable ocean observing. In Proc. AAAI. v

[4] Penberthy, S., and Weld, D. (1994). Temporal Planning with Continuous Change. In Proc. AAAI. v

[5] Vallati, M.; Fawcett, C.; Gerevini, A.; Hoos, H.; Saetti, A.; (2011).ParLPG: Generating Domain-Specific Planners through Automatic Parameter Configuration in LPG.7th International Planning Competition 2011.

[6] Long, D., and Fox, M. (2003). The 3rd international planning competition: Results and analysis. JAIR 20:1–59.

[7] McDermott, D. (1982). A temporal logic for reasoning about processes and plans. Cognitive Science 6:101–155.

[8] Smith, D., and Weld, D. (1999). Temporal planning with mutual exclusion reasoning. In Proceedings of IJCAI-99,Stockholm, 326–337.

[9] Vila, L. (1994). A survey on temporal reasoning in artificial intelligence. AI Communications 7:4–28.

[10] Bacchus, F., and Kabanza, F. (2000). Using temporal logic to express search control knowledge for planning. Artificial Intelligence 116(1-2):123–191.